UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

A Hybrid Intelligent System for Diagnosing and Solving Financial Problems

tese submetida à Universidade Federal de Santa Catarina para a obtenção do grau de Doutor em Engenharia.

Roberto C. S. Pacheco

Florianópolis

Santa Catarina-Brazil

Março de 1996

ROBERTO C. S. PACHECO

A Hybrid Intelligent System for Diagnosing and Solving Financial Problems

Esta tese foi julgada adequada para a obtenção título de "Doutor em Engenharia", especialidade Engenharia de Produção e aprovada em sua forma final pelo Programa de Pós-graduação.

Ricardo Ricardo M. Barcia, PhD

Coordenador do Programa de Pós-Graduação em Engenharia de Produção

Banca Examinadora:

Ricardo M. Barcia, Ph.D, advisor

Suresh K. Khator, Ph.D

Abraham Kandel, Ph.D

Rogério Cid Bastos, Dr..

Fernando Ostuni Gauthier, Dr.

Luis Fernando Jacinto Maia, Dr.

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

A Hybrid Intelligent System for Diagnosing and Solving Financial Problems

dissertation submitted to the Federal University of Santa Catarina to fulfill the requirements for the degree of Doctor in Engineering with emphasis in Production Engineering.

Roberto C. S. Pacheco

Florianópolis

Santa Catarina-Brazil

March, 1996

To Andrea

Acknowledgments

It is always difficult to appropriately recognize all the people we are in debt. Nevertheless, I would like to acknowledge some who have supported me immensely in my dissertation.

First and foremost, I was fortunate to have in Prof. Ricardo Barcia more than an adviser. He has always been a precious friend and source of motivation. Among the things I have learned from him, perhaps his constant search for new professional and personal horizons has influenced me the most. I cannot thank him enough for keeping the standards of this dissertation high while stimulating and fortifying me during the arid times.

I am in debt with the Brazilian citizens who have supported me financially during these years of study. I would like to thank the governmental agency CNPq for its support. There I also found a friend, Carlos Pittaluga, who has helped us to see how to recompense this support.

It would have been impossible to conclude this work without the support of Dr. Suresh Khator, my co-advisor, during my two years in Tampa, Florida, at the University of South Florida. His guidance was priceless not only for helping me in my research but also for reducing the anxiety that moving to another country entails. I also thank Dr. Khator's family for the time we had been together.

I would like to thank Dr. Paul Givens, the chairman of the Department of Industrial and Management Systems. The Brazilian group has always agreed that Dr. Givens is, most of all, a gentleman who made our state at the department a pleasant experience.

During the last year in Tampa, I had the opportunity to meet Dr. Abraham Kandel, the chairman of the Computer Science Department, and an eminent researcher in the Artificial Intelligence community. I am grateful to Dr. Kandel for the opportunity to work together in some publications, and foremost for having coming to Florianópolis for our defenses.

There are several friends who have discussed and suggested improvements to this work. Among them, one did much more than that: Alejandro's work and friendship became an essential part of this dissertation. His knowledge in Finance and endless willingness to learn made our team work a truly pleasant experience.

I would like to thank also the colleagues of the Applied Artificial Intelligence group of the Production Engineering Department of the Federal University of Santa Catarina. Vini, Tite, Fernando, Paulo, Gauthier, Oscar, André, and Paulo Sérgio. I have a close "thank you" to Lori and Rosina for sharing several moments in USA and for all good suggestions to my work.

Last but definitely not least I would like to thank my family for having supported my decisions and for having always considered education a priceless asset.

Abstract

Our main purpose in this dissertation is to develop a system to diagnose and indicate solutions to financial health problems of small and medium firms (SMF). Although monitoring and adjusting financial problems play a central role in the firm's performance, usually SMF face difficulties in these tasks for lacking human resources and for having incapacity to afford a consultant.

The closest types of systems available in the literature are the bankruptcy prediction, the credit analysis and the auditing models. Bankruptcy models do not work for the purpose of financial health evaluation because, rather than looking for causes and corrections to deviations, they intend to foresee the death or life of the firm. Credit analysis models are developed to creditors interested only in the safety of their investments. Auditing systems are limited to ratio analysis with general comments about the financial condition of a firm. After all, the critical aspect of offering practical solutions to the firm remains open. There is need for a financial advisor that helps the manager to make financial decisions that lead to a long-run profitability and success of the firm.

A study of financial statement analysis shown that there are two different reasoning processes participating of the solution: *inductive* and *deductive*. We implemented a *hybrid intelligent* (neuro-fuzzy-symbolic) system to combine the inductive and deductive reasoning processes. The inductive reasoning was modeled by the connectionist module while the deduction was implemented through a fuzzy expert system.

1

Introduction

In 1991, the Artificial Intelligence group of the Production Engineering Department of the Federal University of Santa Catarina started to work on a project of building intelligent systems capable of helping small and medium firms (SMF) on their most critical tasks. The relevance of this kind of businesses to the Brazilian economy can be understood by a report which states that small business activity is responsible for 40% of the total Brazilian Gross National Product (GMT) [CHER90]. Particularly in the state of Santa Catarina a research done by the Production Engineering department indicated that financial management and production planning were considered the most critical areas by the small businesses consulted [BATA90].

The work has resulted in five graduate thesis and seven doctorate dissertations related to different aspects of the financial management and the production planning of firms.

In this broader picture, this dissertation is concerned with the evaluation and correction of financial problems of SMFs through computational models. This task requires an adequate modeling and representation of the financial knowledge. The knowledge modeling has to be accurate without loosing the possibility of being represented computationally. On the other hand, once the knowledge has been elucidated, one has to identify an adequate computational model that represents the different aspects of the problem. Our main purpose is concerned with the last task.

1.1. Importance

The overall performance of a firm depends on the balance between liquidity and profitability. Monitoring financial problems must be a continuous task. Even a very profitable firm can hide factors that undermine its insolvency. The monitoring task identifies factors that deviate from the trend in time for making adjustments.

Usually a firm has two alternatives when monitoring and solving problematic activities: do it by itself or hire a consultant. While the first choice requires human resources, hiring a consultant can be a solution only if the firm can afford it. Generally both choices can be a problem to small companies. Besides, even when one of these alternatives is available, the solution may be arrived only after a slow process rendering it to be useless.

One approach for this problem is the development of intelligent systems that not only analyze financial problems but also suggest solutions. The aggregation of both tasks involves deductive and intuitive reasoning which justifies the use of more than one Artificial Intelligence technologies, that is, a *hybrid intelligent system*. One type of integration involves Fuzzy Expert Systems and Neural Networks (neuro-fuzzy systems). The aim is achieve improvements in the implementation of each and increase the scope of application [MEDS94b].

A bibliographical research revealed that there are three kinds of intelligent systems dedicated to financial health analysis: bankruptcy prediction (*e.g.* [LACH91] and [WILS94]), credit analysis (*e.g.,* [BARK90a]) and auditing (*e.g.,* [BLOC90] and [MUI90]).

Bankruptcy models do not work for the purpose of financial health evaluation. Rather than looking for causes and corrections to deviations, they intend to foresee the death or life of the firm. Credit analysis models are developed to creditors interested only in the safety of their investments. Auditing systems are limited to ratio analysis with general comments about the financial condition of a firm. After all, the critical aspect of offering practical solutions to the firm remains open. There is need for a financial advisor that helps the manager to make financial decisions that lead to a long-run profitability and success of the firm.

We propose a hybrid intelligent system that diagnoses and indicates solutions to financial problems of small firms. The system integrates a Neural Network (during the diagnostic phase) and a Fuzzy Expert System (during the solution phase).

1.2. Organization of the Work

This work is organized in four chapters. Chapters 2 and 3 are related to the technologies involved in the application of this dissertation. Both chapters bring the foundations and applications of the technologies used in this work. In Chapter 4 we focus on the application of hybrid technologies proposed in this work. The organization is the following:

Chapter 2 - Hybrid Intelligent Systems

In this chapter we describe the technology of hybrid intelligent systems. Particularly, we concentrate on the most common form of hybrid systems: the integration between neuron networks and expert systems. We analyze both technologies from a historical point of view. We also compare the advantages and shortcomings of each technology. Finally, we present the definition and architectures of hybrid intelligent system as well as their application in several areas of Engineering and Finance.

Chapter 3 - Artificial Intelligence Technologies

In this chapter we discuss the foundations of the Artificial Intelligence technologies used in the work. The goal here is to establish the theoretical grounds for the neural network architectures implemented in the diagnosis phase of the system (RBF and Backpropagation) and for the fuzzy expert system implemented in the financial decision module. First, we address the foundations of classical expert systems, Fuzzy Set Theory and Fuzzy Logic related to the deductive module of the application. Then we discuss both neural models, RBF and Backpropagation, which appear as alternatives to the inductive module in our application.

Chapter 4 - Application: Financial Health of Small Retail Firms

In this chapter we discuss the definition and relevance of the problem in the field of Finance. We explain the basis for the choice and generation of the financial variables involved in the problem. The discussion is based on the findings of Martins about the more suitable tools in Finance to diagnose and indicate solutions to financial problems [MART96]. After analyzing the problem under a financial point of view, we discuss the architecture and component parts of the system, concluding with a complete example of application, from the ratio analysis (diagnosis) to the indication of actions to be taken by the firm regarding the problems identified.

Chapter 5 - Conclusions and Future Work

We believe that this dissertation is an opening field for future work regarding both intelligent financial systems and hybrid techniques. In the last Chapter we discuss several further developments to this work as well as its main conclusions regarding computational financial consulting and hybrid system implementation.

2

Hybrid Systems

2.1. Introduction

Artificial Intelligence (AI) began in 1956 with a historical conference in Dartmouth, New Hampshire. John McCarthy (who later invented LISP) defined AI as the field of Computer Science dedicated to the study and modeling of human intelligence. The researchers at the Dartmouth Conference (Marvin Minsky, Allen Newell, Herbert Simon, among others) believed that intelligent behavior should be modeled at *functional* rather than *physiological* level. This paradigm is called the *macro view* of intelligence according to which the human brain can be thought of a black box whose interior is not relevant to its intelligent behavior. Philosophers, logicians, decision analysts and psychologists are among the researchers of this paradigm. The main concern is the study of the thinking process and the mechanism of deduction of hypothesis from confirmed evidences. *Expert Systems* are the most significant technique developed under the macro view of intelligence.

At the same time that the Dartmouth delegates were developing the grounds to the macro view, another group of researchers was following a different approach to model intelligence. Neurologists and physicians among others analyzed intelligence according to a *micro view*. They investigated intelligent behavior by describing chemical and physical events taking place at the brain during the thinking process. Rather than describing intelligence according to a *functional* view, they modeled intelligent behavior by studying it at the *neuro-physiological* level. The main concerns were to understand the brain operations, how information is stored and transferred, how humans remember or forget facts and mainly, how humans learn [ZAHE93]. *Neural Networks* are the principal technique developed according to this paradigm.

During almost thirty years these two distinct paradigms were developed independently or, sometimes, competitively (e.g., [MINS69]). The remaining question is: which approach should be chosen? Like many others, Marvin Minsky, an AI pioneer, believes that both paradigms should be applied. While symbolic models have indicated how to make machines solve problems by resembling reasoning, neural models have enlightened the role of brain cells in the process. The problem is that both models are very far from the complete understanding of the thinking process and the neural system. The two approaches form the extremes in our gap of knowledge [MINS88]. The solution seems to lie in reducing this gap by integrating both visions of intelligence modeling.

If the aim is to model natural intelligence, by choosing either symbolic AI or neural networks, one becomes a *reductionist*. Callataÿ has described reductionists as scientists with a common background of knowledge and whose findings are bricks to a robust science. Reductionists face difficulties in observing the aspects of a problem that are outside their domain. Natural intelligence is a result of a such complex system that it is probably a misconception to assume that it can be described by any unified theory [CALL92].

In the last years, researchers from both views decided to take a different direction. New models have been proposed based not on a single paradigm but on the combination of both. While symbolic systems indicated how to implement deductive reasoning, connectionist models showed the role of neural cells in other aspects of intelligent behavior. The main goal became to combine both paradigms in search of a more accurate model of intelligence.

In this work, we assume the general and new concept of Artificial Intelligence rather than the one adopted by the Dartmouth Conference. In the old view, although defined as the field of Computer Science dedicated to model human intelligence, AI was associated exclusively to the symbolic approach. According to the modern definition, any method that models some aspect of human intelligence is considered an AI tool (e.g., neural systems, evolutionary computation, genetic algorithms, etc.).

2.2. Definition and Hybrid System Architectures

2.2.1. Definition

Intelligent Hybrid Systems are defined as models and computational programs based on more than one Artificial Intelligence technology. In Figure 2-1 we represent a schematic view of the integration of some AI techniques that can be combined in hybrid architectures. Different aspects of intelligent behavior can be modeled. Machine Learning and Neural Networks model intuitive and inductive reasoning; Evolutionary Systems model the adaptive behavior; Fuzzy and traditional Expert Systems model deductive reasoning; and Case Based Reasoning combine deduction and experience. The input data in a hybrid system may be treated for some procedural preprocessing method and then passed to the system modules. These modules interact during the solving process or prepare the data to future processing by other module. The result can be sent to another procedural processing to prepare the system output. Depending upon the problem complexity, this output may be entrance to another hybrid system which works in a similar way.

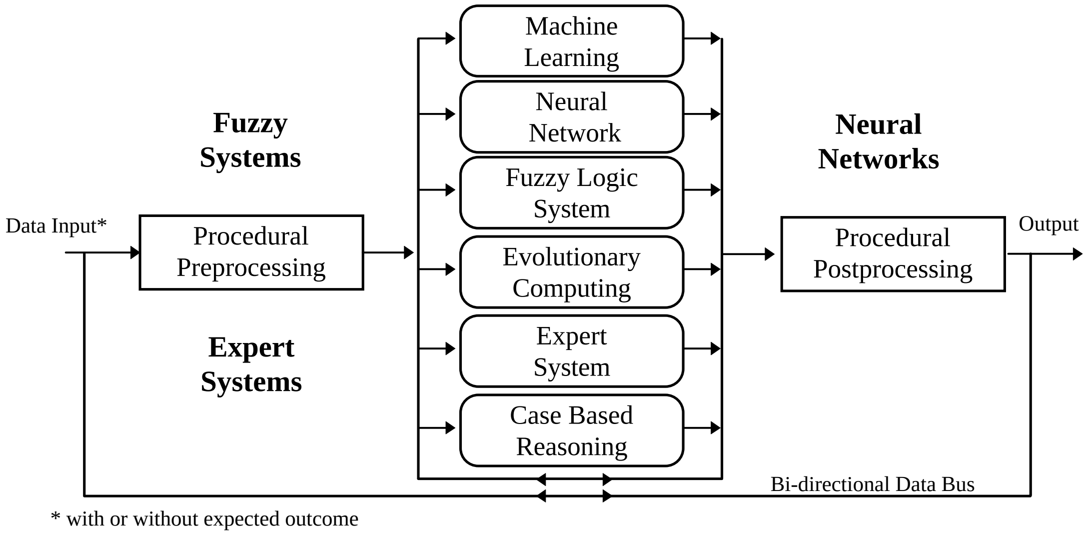


Figure 2-1: Components of Intelligent Hybrid Systems (adapted from [SCHW91] e [KAND92b]).

In this dissertation, the integration model involves the combination of a fuzzy expert system with a neural network module. In the following section, we discuss the comparisons between these two AI technologies and the benefits of their integration.

2.2.2. Expert Systems x Neural Networks

Hybrid Systems integrating neural and symbolic models (*neuro*-symbolic systems) aim two goals: conciliate the advantages of both techniques in an integrated model more powerful than its parts alone; and to overcome the deficiencies of one approach with the strengths of the other. Hence, one has to know the advantages and disadvantages of each technique before integrating them. Table 2-1 depicts some advantages of each technology expected to be simultaneously in the neuro-symbolic hybrid system.

Table 2-1: Expected Advantages of Hybrid Systems Integrating NN e SE

Feature	Description	Found in
Learning	The system can learn from its own experiences.	*Neural Networks*
Integration of different sources of knowledge	Different sources of knowledge may be combined (e.g., vision, hearing, touch, etc.).	*Neural Networks*
Fault Tolerance	Failures in individual parts of the system do not affect its overall performance.	*Neural Networks*
Generalization	An output is guaranteed even when the system was not prepared to deal problems as the one being treated.	*Neural Networks*
Explanation	The system is able to explain its answers or the need for the data being required.	*Expert Systems*
Symbolic Reasoning	The system accepts, processes and presents symbolic data (natural language).	*Expert Systems*
Parallel Processing	It can be implemented in parallel processing architectures.	*Neural Networks*

In the next section we discuss different forms of integration between ES and NN.

2.2.3. Hybrid System Architectures

There are several architectures for integrating neural and symbolic models. They have been classified according to the task executed by each module [RICH90], to the functionality, processing architecture and communication requirements [GOON95], or according to the architecture used to implement the system ([MEDS92] and [MEDS94]).

In Table 2-2 we summarize the Medsker and Bailey's architecture classification [MEDS92]. Based on the implementation architecture, the classification goes from the total independence to the complete synergy between the modules. Medsker describes this models as the following:

Table 2-2: Architectures for ES and NN Integration [MEDS94].

Architecture	Description	Example
Stand Alone SE RN	There is no integration or communication between the modules. They are implemented independently and used to solve the same problem.	Comparison of different diagnostics in computer repair.
Transformational SE RN	Independent modules that do not interact. The system is built as one model and rebuild as the other.	Model marketing decision aid. A NN is built to identify trends and relationships within the data and it used as the basis to build an expert system to assist marketing researchers in allocating advertising
Loose Coupling SE RN	ES and NN form independent modules that interact via data files.	Forecasting of workforce utilization. The NN predicts the workforce and the ES allocates the tasks.
Tight Coupling SE RN	ES and NN are separate modules that communicate via memory resident data structures.	Forecasting stock prices (NN) and consequent definition of appropriate strategy (ES).
Full Integration SE RN	ES and NN form a single structure, sharing data and knowledge representation.	Object identification based on feature data received from sensors (NN) and environmental data (ES).

a) Stand-Alone

Stand-Alone architectures are composed by independent modules without any integration between the parts. Although these models are not an alternative to hybrid solutions, they have some advantages. First, they are a direct means of comparing solutions offered by both techniques for a same problem. Second, the implementation of a module after the other allows the validation of the first system. Finally, running the modules simultaneously allows a loose approximation of the integration.

b) Transformational

Transformational models are similar to stand-alone architectures regarding the independence between the modules. In transformational models the system begins as one model (ES or NN) and ends up as the other. There are two types of transformational models: expert systems that became neural networks and vice versa. The choice for one or the other is based on the features of each technique (Table 2-1) and on the final solution required. For instance, certain system may require real time response (e.g., control systems) but the solution may be based on deductive reasoning. The knowledge engineer may develop an expert system and transform it to a neural network in order to use a parallel architecture in the final solution. The limitations of transformational models include the absence of automated means of transforming one technique to the other and the difficulty in maintain both modules when new features are added to the solution.

Medsker and Bailey [MEDS92] present the architectures of neuro-symbolic systems as a pyramid with the stand-alone and transformation models as the basis. Rather than hybrid architectures the authors consider these models two forms of relating the two intelligent techniques. In these models there is no integration between the modules. They are alternative solutions to the same problem.

c) Loosely-coupled

The first hybrid architecture is called *loosely-coupled*. In these models expert system and neural network are independent and separate modules that communicate via data files. Both modules can be pre, pos or co-processors in relation to the other. In this dissertation we implement the *neuro-pre-processing* model. In this model the solution begins with the NN that processes the data (fusion, error reduction, object identification, pattern recognition, etc.) and stores the solution in file. This is the input to the expert system module to the task of classification, problem solving, scenario analysis, etc. The major advantage of loosely-coupled is that they are easy to implement (each module can be developed separately in several shells available commercially) and maintain (due to the simple interface between the modules). On the other hand, this architecture is slower in the operation and may have redundancy in the independent developments of each module (identical data may be considered independently).

d) Tightly-Coupled

The next level of integration is the *tight coupling* model. The only difference is how the communication between the modules takes place. In this modules, NN and ES communicate through memory resident data structures. Although the solution in the user point of view is the same as in loosely-coupled models, there are several functional consequences due to the communication in the main memory. Due to this functional feature, tightly-coupled architectures are more adequate to embedded systems, blackboard architectures and real time cooperative systems. On the other hand, these systems have the same problems of redundancy and maintenance of loosely-coupled systems.

e) Fully Integrated

The last level of integration is the *fully-integrated* architecture. NN and ES share data and knowledge representation and the communication between them is accomplished due to the dual nature (neuro-symbolic) of the structures. The reasoning mechanism is implemented according to a cooperative scheme or with one module playing the role of the inference controller. There are many forms of full integration, including the connectionist systems [GALL88], utilization of I/O neurons, subsymbolic to symbolic connectivity and integrated control mechanisms.

The key to the choice of the hybrid architecture is the problem to be solved. As a general rule, the integration is suitable whenever the solution involves both deductive and inductive reasoning. The choice for a specific architecture depends upon the way the intelligent agents cooperate in the process, upon the computational resources available and upon the kind of solution required.

2.3. Applications of Hybrid Systems

Even being relatively new, hybrid neuro-symbolic systems have several applications, some in commercial environments. These include systems that require fault tolerance, generalization, implicit and explicit reasoning, incremental learning and flexible architectures [LIEB93]. Table 2-3 has some examples of applications of neuro-symbolic systems in several areas of engineering, medical and company diagnostic. Hybrid systems are a current trend in Artificial Intelligence ([HAYE94], [MEDS94], [MEDS95], [GOON95], [MUNA94] and [ZAHE93]). Particularly regarding neuro-symbolic systems, one can expect the dissemination in several areas, especially with the development of hybrid shells that integrating different AI technologies under the same environment. Nevertheless, the commercial application of hybrid systems in large scale depends on research in several topics, including the study of unified architectures rather than hybrid solutions and the development of formal knowledge representation models for neural networks [HUAN95]. The increasing number of publications in this area (*e.g.,* [KAND92b], [HONA94], [SUN95], [MEDS95]) reveals that several researchers are working in hybrid systems and one can expect solutions to these and other open issues.

Table 2-3: Examples of Applications of Hybrid Systems Involving NN e ES.

Year	Application	Author
1988	Medical Diagnostic	Gallant [GALL88]
1988	Work force forecasting in maintenance of workstations.	Hanson and Brekke [HANS88]
1990	Dispatching delivery trucks weight and volume constraints to minimize the number of trucks required and the total miles traveled.	Bigus and Goolsbey [BIGU90]
1990	Preclassification of DNA samples in HIV studies.	Benachenhou *et al.* [BENA90]
1990	Financial diagnostic of firms to evaluate the probability of loan.	Barker [BARK90a] and [BARK90b]
1991	Mapping of acoustic signals to symbols (allowing symbolic reasoning).	Hendler and Dickens [HEND91]
1991	Financial Diagnostic of Firms.	Nottola *et. al.* [NOTT91]
1991	Evaluation of hypotheses in problem solving.	Gutknecht *et al.* [GUTK91]
1992	Production Planning in manufacturing flexible systems.	Rabelo and Alptekin[RABE92]
1992	Quality Control (corrosion prediction)	Rosen and Silverman [ROSE92]
1993	Diagnostic of faults and performance control in telecommunication systems.	Senjen, *et al.* [SENJ93]
1993	Modeling and monitoring of industrial processes.	Markos *et al.* [MARK93]
1994	Forecasting performance of stock prices.	Yoon et. al. [YOON94]

3

Artificial Intelligence Technologies

3.1. Introduction

In this chapter we present a review of the two Artificial Intelligence technologies considered in this dissertation: *fuzzy expert systems* and *neural networks*. First, we present the main concepts of expert systems and the issue of uncertainty treatment in these systems. Among various alternatives available for dealing with uncertainty, the technique chosen is based on Fuzzy Models. The essential elements of Fuzzy Set Theory related to uncertainty treatment and the two models of reasoning in Fuzzy Logic (Approximate and Possibilistic) are described. The fourth section is dedicated to issues related to artificial neural networks. In this section there is a more detailed description of the *Backpropagation* and *Radial Basis Function* algorithms, the models applied in the hybrid system implemented.

3.2. Expert Systems

An *expert system* is an intelligent computer program developed to emulate the reasoning process of an *expert* in a specific domain. An expert is someone who can solve a problem which most of the people can not for lack of training or knowledge. More than knowledge, an expert system encodes *expertise*. Knowledge can also be acquired from other sources such as books, periodicals or other media. When the knowledge in the system is not acquired from a human, the system is called *knowledge-based system*. More recently, expert systems and knowledge-based systems have become synonymous. They also became a paradigm to conventional algorithmic programming [GIAR94].

Contrary to traditional systems (based on algorithmic programming), experts systems have built-in facilities that increase flexibility and efficiency. The most important are the possibilities of creating rules, gathering facts, and making logical decisions under imprecision or even in absence of information. In a standard program, the search is a procedural method based on the previous knowledge codified in the system. When new knowledge emerges, it is necessary to rewrite the code. On the other hand, an expert system can retrieve new facts and rules and use them without changes in the search strategy.

3.2.1. The Building Process of Expert Systems

The process of designing and building expert systems is iterative and it is called *Knowledge Engineering.* Figure 3-1 gives an overview of the entire process of building an expert system. The design starts with the *Knowledge Acquisition*, the process of obtaining the knowledge required. The knowledge engineer gathers the information by consulting the expert and references. This is a critical activity in the process (known as the “bottleneck”) and involves mainly the identification, assessment, and familiarization with the problem [SCOT91]. The knowledge codification phase is called *Knowledge Representation*. Several methods are available [RING88] and the most important knowledge representation techniques are: *semantic networks* [QUIL68], [SOWA91], *rules* [BUCH84], [JACO86], and *frames* [MINS75] (and more recently its derivation: *object-oriented methods* [STEF86], [BONC93], [KAIN94]).

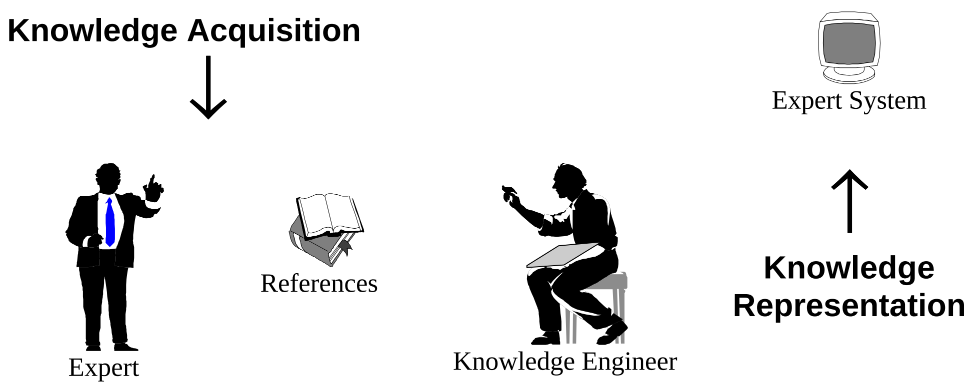


Figure 3-1: Buiding Process of an Expert System

3.2.2. Rule-Based Systems

Rules are the most common form of representing knowledge. An expert system whose knowledge is codified in rules is called *Rule-Based System*. The knowledge in this kind of system is represented by *production rules* in the following form:

IF antecedent(s) THEN consequent(s)

where both antecedent (or “premise”) and consequent (or “conclusion”) are logical propositions or clauses. For example: “IF the temperature exceeds 120^{o}F THEN turn off the engine". Multiple antecedents (or consequents) are connected by the logical connectives "AND" or "OR". For instance: "IF (the temperature exceeds 120^{o}F) OR (the oil level is below the minimum) THEN turn off the engine".

A rule-based system reaches its conclusions either by inferring the consequents whose antecedents have been proved (*forward chaining*) or by assuming consequents and proving them by confirming the antecedents (*backward chaining*) [WATE78]. These inference mechanisms where answers are drawn based on rules and facts form the *deductive reasoning*, an inferential system based on methods of Classical Logic.

Originally, deductive reasoning was implemented without uncertainty in the clauses or in the validity of the rules. Unfortunately, human reasoning often does not follow this deterministic approach. Indeed, dealing with uncertainty is a must to an accurate representation of human reasoning.

3.2.3. Uncertainty in Expert Systems

During the knowledge acquisition, the knowledge engineer must also check whether the problem requires inexact reasoning in its inference or in the treatment of the information. In other words, the knowledge engineer must check how the expert system should make inexact conclusions and how it should use these conclusions [SCOT91]. This is usually called *Uncertainty Treatment* [SHOR76], [KANA86], [BOUC93].

While the first Knowledge Acquisition and Knowledge Representation studies emerged during the early developments of expert systems [FEIG92], the management of uncertainty came only in the seventies. The AI scientists realized that as long as an expert system intended to emulate the human reasoning, it should be able to deal with uncertainty. The sources of uncertainty include:

- vagueness of human concepts,
- incomplete or unreliable information,
- disagreement between different sources of knowledge (experts), and
- partial matching between facts and evidences.

The challenge to AI researchers has been to empower the systems with the capability of dealing with this variety of uncertainty[1]. In the last decades several theories have been proposed as alternatives to implement this task, including:

- Probabilistic methods (*e.g.,* [DUDA76], [PEAR86], [PEAR88], [NEAP90], [HECK95]);
- Certainty Factors (*e.g.,* [SHOR76], [BUCH84], [HECK86]);
- Dempster-Shafer Theory (*e.g.,* [SHAF76], [YAGE94b]);
- Fuzzy Models (*e.g.,* [ZADE83], [KAND92], [KAND96]).

The first three models address only uncertainty and ignorance while Fuzzy Models can deal with vagueness of natural language terms. In this dissertation, the modeling of imprecision is accomplished by fuzzy models. The reasons lie in the nature of Financial Statement Analysis, where rules are vague (described by vague terms) and uncertain (dependent on the financial environment). Hence, there is need to address *Fuzzy Expert Systems*.

3.3. Fuzzy Expert Systems

Fuzzy Expert Systems (FES) have the same components of classical ES but use Fuzzy Set Theory to model uncertainty of attributes and Fuzzy Logic to implement the inference. Fuzzy Set Theory and Fuzzy Logic provide a theoretical ground to FES where the uncertainty management by strength of belief (modeled by other methods) is a subcase. Fuzzy expert systems admit fuzziness to express uncertainty in antecedents, consequents or even in the logical relation between them (rule). By fuzziness we mean the imprecision of grouping elements into classes that do not have sharp boundaries.

There are two theoretical frameworks in Fuzzy Set theory to model the logical mechanism of inference of fuzzy expert systems: *Approximate Reasoning* [ZADE79] (or *Fuzzy Reasoning* [GAIN76]) and *Possibilistic Reasoning* ([ZADE78] and [DUBO88]). These theories are based on Fuzzy Logic whose foundations lie on *Fuzzy Set Theory* [ZADE75].

---

[1] Although the term *uncertainty* has coined this discipline, it should be noticed that a more appropriate name would be *imprecision* as the general term for describing the variety. Uncertainty is also the specific kind of imprecision where the degree of true in the information is not totally certain. This does not include imprecisions such as *vagueness* of natural language terms, *ignorance* and *ambiguity* of information [KAND96].

In Approximate Reasoning, the semantic of fuzzy sets is related to the concept of the linguistic term described by the set. Vague terms such as “medium” and “moderate” are modeled by fuzzy sets and can be related by fuzzy rules. Fuzzy Set Theory is the foundation for representing these elements while the inference mechanism is based on Fuzzy Logic. This forms the essence of Approximate Reasoning. In this theory, several forms of inference systems can be identified, according to the conception of a fuzzy rule (*conjunction* or *implication*) and the definition of union and intersection operators. Particularly, when the answer has to be a real number, there is need for *defuzzification* process which translates the fuzzy set deduced from the inference. Defuzzification is typical in Fuzzy Control (*e.g.,* [KAND94] and [YAGE95]), one of the principal areas where Approximate Reasoning is applied successfully [KAND96].

When uncertainty is involved, the fuzzy sets are seen as constraints to the actual values of variables (*i.e.,* descriptions of imprecise knowledge). In this framework, fuzzy sets are *possibility distributions*. Possibility Theory and Fuzzy Logic establish the elements of *Possibilistic Reasoning*, the theoretical basis for fuzzy systems whose inference admits uncertainty. The relationship between Fuzzy Set Theory and Probability Theory can be done only within the Possibilistic view of a fuzzy set.

For the reasons explained in Chapter 4, in this dissertation only Approximate Reasoning is applied. In the remaining sections we revise the basic concepts of these theories in the context of fuzzy rule-based systems used in the application of this dissertation.

3.3.1. Fuzzy Set Theory

In Classical Set Theory, an element x either belongs to or does not belong to a set A. The notion of membership is crisp (dichotomic). A membership value (*i.e., characteristic function* value) is either 0 or 1 (within the set {0,1}). Consequently, there is a lack of means to represent fuzziness. In other words, an ordinary set is inappropriate to translate into computational form concepts such as “small,” “hot” and “tall” because elements match these classifications with different grades rather than the single pair of full (1) and null (0) memberships.

For example, a concept such as “small file” in computers, could be represented by the crisp set: {x = “small file” iff x ≤ 100K}. But how about 101K files? Are they so different from 100K files? Despite our intuition, the previous representation would classify this two similar kinds of files in different categories. This is what happens when the classifications have sharp boundaries. Somehow, one has to differentiate the degree with which the elements belong to a set. An 1K file is much more smaller than a 99K file and such distinction can not be ignored in a computational representation of the concept "small files". Fuzzy Set Theory was created with the purpose of overcoming the misrepresentation of human concepts as crisp sets.

3.3.2. Fuzzy Set

In 1965, Lotfi A. Zadeh coined the notion of a *Fuzzy Set* as a "class of objects with a continuum of grades of membership" [ZADE65]. Zadeh generalized the idea of a crisp set by extending the interval of values of the characteristic function from the set {1,0} to the range [0,1]. Given a space of objects (elements) $X = \{x\}$, a fuzzy set A is a set of ordered pairs defined by:

$$A = \{x, \mu_A(x) \mid x \in X \text{ and } \mu_A(x) \rightarrow [0,1]\} \qquad (3\text{-}1)$$

where $\mu_A(x)$ is called *membership grade*. Every fuzzy set establishes a mapping from the universe of discourse to the interval [0,1] called *membership function*. This mapping represents the notion of partial belonging of an element to a class. Therefore, the class has non-rigid boundaries and it is defined not only by its elements but also by the associated membership grades.

When defining a fuzzy set, it is common to use a *normalized fuzzy set*, that is: $\exists w,\ \mu_A(w)=1$.

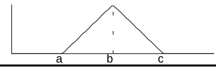
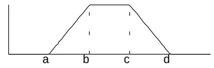
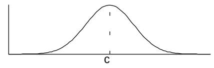
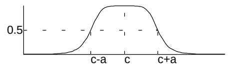
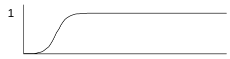
A membership function can be either discrete or continuos. The notation of the fuzzy set in each case is the following:

$$\text{discrete:} \qquad A = \Sigma_U\ \mu_A(u)\ /u \qquad (3\text{-}2)$$

$$\text{continuos:} \qquad A = \int_U\ \mu_A(u)\ /u \qquad (3\text{-}3)$$

Notice that both symbols 'Σ' and '$\int$' stand for the union of all elements in the set. They do not represent algebraic sums as in Calculus. In most cases, the membership function appears in the continuos form. There are several alternatives: triangular, trapezoidal, gaussian, bell and sigmoidal membership functions are the most commonly used. Table 3-1 presents them as parametric functions [JANG95].

Table 3-1: Parametric formulas of Common Continuos Membership Functions.

Function	Formula	Graphic
triangular	$\mu_A(x)$ = max [min{(x-a)/(b-a);(c-x)/(c-b)};0]	
trapezoidal	$\mu_A(x)$ = max [min{(x-a)/(b-a);1;(d-x)/(d-c)};0]	
gaussian	$\mu_A(x) = e^{-((x-c)/\sigma)^2}$	
bell	$\mu_A(x) = (1 + \lvert (x-c)/a \rvert^{2b})^{-1}$	
sigmoidal	$\mu_A(x) = (1 + exp(-a(x-c))^{-1}$	

3.3.3. Essential Operations on Fuzzy Sets

In the classical theory of ordinary sets, an operation is a relation between sets that yields another set. Operations on sets are relevant to expert systems because the inference process is based on the processing of logical connectives (propositional calculus) which have equivalent quantifiers in Classical Set Theory. For instance, given the rule `IF (A AND B) THEN C`, the true value of `C` is the true value of the conjunction (`A AND B`), that is, the value of the intersection ($A \cap B$). For the same reason, operations on fuzzy sets are crucial to the fuzzy expert system inference. The extension of set operations to fuzzy sets is not obvious. The three basic operations, *union*, *intersection*, and *complement* of fuzzy sets were originally defined by Zadeh as [ZADE65]:

union:	A or B	$\Rightarrow$	$A \cup B$	=	$\{x, \max(\mu_A(x); \mu_B(x))\}$	(3-4)
intersection:	A and B	$\Rightarrow$	$A \cap B$	=	$\{x, \min(\mu_A(x); \mu_B(x))\}$	(3-5)
complement:	not A	$\Rightarrow$	$\neg A$	=	$\{x, \mu_{\neg A}(x) \mid \mu_{\neg A}(x) = 1 - \mu_A(x)\}$	(3-6)

where $x \in U$ (universe of discourse)

When the membership grades $\mu_A(x)$ and $\mu_B(x)$ are confined to the set {1,0} (*i.e.,* when *A* and *B* are ordinary sets), (3-4), (3-5) and (3-6) become the classic notions of union, intersection, and complement, respectively. These basic definitions of fuzzy set operations are not only natural but also quite reasonable regarding some desirable assumptions to their behavior [BELL73]. Indeed, maximum and minimum have been extensively and successfully used in practical applications of Fuzzy Set Theory ([TERA94] and [HIRO93]). Nevertheless, they are not the only alternatives to extend the classical operations. Many operators have been proposed to model fuzzy union and fuzzy intersection and different definitions will yield different fuzzy expert system models [KAND96].

The relevance of the investigation of fuzzy set operators is not restricted to the mathematical ground of Fuzzy Set Theory [DUBO85]. In practical situations the choice of appropriate operators depends upon the knowledge about the features of the problem, the alternative operators available and their distinct properties. For instance, in decision-making process, *max-min* operators are not appropriate if the decision criteria are compensable, *i.e.*, the appropriateness of one can balance the inadequacy of the other. In such situations, a better choice would be a compensatory operator such as average or geometric mean [DUBO80]. The choice for *max* and *min* operators in this work is due to the following facts: first, there was explicit intention of meeting the axiomatic framework in which fuzzy unions and fuzzy intersections have been described (*i.e.,* using t-conorms and t-norms); second, we did not find any reason for adopting more restricted in intersections (*i.e.,* for not using the largest fuzzy set produced by the intersection) and more for using more relaxed operator in unions (*i.e.,* for using any other fuzzy union greater than the standard operator). Nevertheless, any other operator could be used based on axiomatic frameworks of Fuzzy Logic (described, for instance, in [KLIR95], Ch. 3)

3.3.4. Approximate Reasoning

Approximate Reasoning [ZADE79] (or *Fuzzy Reasoning* [GAIN76]) can be understood as the process of inferring imprecise conclusions from imprecise premises. This deduction process is “approximate” rather than “exact” because data and implications are described by fuzzy concepts. *Fuzzy Logic* in the narrow sense (*i.e.*, a logical system extended from multi-valued logic [ZADE94]) is the ground theory to Approximate Reasoning. In the following sections, we present the essential elements of Fuzzy Logic in the management of uncertainty in expert systems and the logical framework that underlines the inference process in Approximate Reasoning.

a) Fuzzy Rules

A Fuzzy Rule is an implication between fuzzy propositions (clauses with fuzzy predicates). Examples of such implications are common in natural language: “if the car is noisy, there is a chance for mechanical problems,” “if the apartment is spacious and affordable, let’s buy it,” etc. The first step in modeling fuzzy rules is to identify its logical framework, that is, to specify how a fuzzy implication is logically represented. In fuzzy systems, rules are conceived in two distinct ways: (a) as a *logical implication* between antecedent and consequent; (b) or as the *conjunction* between them. Each view underlies a different method for pursuing the inference in fuzzy systems. In this dissertation (and in the majority of practical applications [MEND95]), fuzzy rules are evaluated as conjunctions of fuzzy predicates. For this reason, this is the only method of fuzzy rule evaluation we discuss here[2].

The rule "IF *x* is *A* THEN *y* is *B*," where *A* and *B* are fuzzy sets, describes a universe of domain delimited by the Cartesian Product of the sets *A* and *B*. The calculus of the truth value of a rule conceived as a conjunction is done by:

$A \rightarrow B = A \cap B$, where the intersection is determined by a *t-norm*.

Originally proposed by Menger [MENG42], *t-norms* and *t-conorms* became suitable concepts for representing pointwise fuzzy-theoretic *intersection* and *union*, respectively [YAGE85]. *Minimum* and *maximum* are examples of these operators and are the only t-norm and t-conorm that meet the property of *idempotency*, that is, when applied to identical elements they yield this same element. This can be advantageous when the sets being conjuncted (or disjuncted) include repetitions. Pattern recognition and multi-criteria decision making are examples of such environments, where identical categories can be displayed with different names [YAGE94].

b) Generalized Modus Ponens

In Fuzzy Logic, the classic rule of inference has to be modified, since facts and implication between them can be fuzzy. A fuzzy rule-based system underlies its deduction mechanism through the so called *generalized modus ponens*:

$$\begin{array}{l} R\text{: } x \text{ is } A \rightarrow y \text{ is } B \\ x \text{ is } A^* \\ \hline \therefore y \text{ is } B^* = A \circ R \end{array} \qquad (3\text{-}7)$$

where *A, B,* A^*, and B^* are fuzzy sets (or, equivalently, fuzzy predicates), *x* and *y* are fuzzy variables, *R* is the fuzzy relation formalizing the fuzzy rule, and '$\circ$' stands for the composition.

There are two main differences between the generalized modus ponens and its classical version: A^* and *A* are not necessarily identical (*e.g.,* A^* is "close" to *A*); and the predicates *A, B,* and A^* are not required to be crisp. In Fuzzy Logic, crisp

[2] Detailed discussions about the logical framework of fuzzy rules can be found in [KANDE95], [DUBO84] and [YAGE83].

predicates are only subcases. When A and A^* are crisp and identical, B^* is equal to B, that is, the generalized becomes the classical modus ponens [ZADE73].

In rules with multiple antecedents, the inference includes the operations of conjunction (for connective AND) or disjunction (for connective OR) before the conclusion of B^*. In the general case, the inference mechanism is applied to several rules, generating many conclusions B_i^*. The final conclusion of the system is the disjunction of all B_i^*.

The inference process based on the generalized modus ponens is not unique. It depends upon the representation of the fuzzy rule and on the operations over the fuzzy sets. Therefore, the generalized modus ponens describes the general syllogism that supports the inference in fuzzy systems but does not establish a unique calculus for the inference. In our application we decided to adopt the most used representation of fuzzy rules (fuzzy conjunction) and t-norm/t-conorm definitions (min and max).

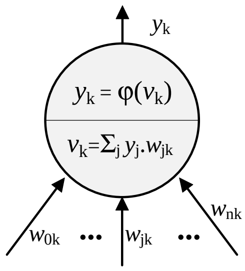
3.4. Neural Networks

Neural networks are computational structures based on parallel distributed processing units (neurons) organized as directed graphs with a learning algorithm that implements knowledge (stored in the connections) from a set of sample data. Therefore, neural networks are *inductive* information processing models, that is, they form general patterns and rules from raw data and experience. In the literature, neural networks are also referred to as *connectionist networks* (models), *parallel distributed* processors, and *neurocomputers* [HAYK94]. The neural network design is inspired in the way the human brain physically works, that is, as a net of neurons distributed in parallel whose electronic activations are responsible for the recovery of specific knowledge. Regarding the two abstractions of natural intelligence discussed in Chapter 2, neural networks are micro view models.

In the following sections we describe briefly the neural network elements related to this dissertation. In the following, we review the kinds of cells, architectures and learning methods of neural networks. The aim is to specify how the models implemented in the application described in Chapter 4 are classified in connectionist theory.

3.4.1. Artificial Neurons

The most basic element of a neural network is the *artificial neuron* (cell). An artificial neuron can be seen as a logical structure of a natural neuron [AMIT89]. Figure 3-2 is a schematic view of an artificial neuron u_k.



k	- index of the artificial neuron being activated
n	- total of connections to u_k
y_j	- output of the jth artificial neuron connected to u_k
w_{jk}	- weight of the connection between neurons *j* and *k*.
w_{0k}	- bias
v_k	- weighted sum of the input signals in *k*
φ	- activation function or neuron *k*'s output.
y_k	- neuron *k*'s output

Figure 3-2: Artificial Neuron - Basic Processing.

Three basic elements describe an artificial neuron [HAYK94]:

1. the arrows represent the *synapses* or connecting links to preceding neurons in the net. Each connection has a *weight* w_{jk} representing the *strength* of the signal sent by neuron u_j to the neuron u_k. The weight w_{jk} is positive if the synapse is excitatory and negative if the synapse is inhibitory. Usually, the *bias* (an increase in the neuron output) is represented by a pseudo-connection w_{0k} and by a virtual neuron u_o whose activation level is constant and equal to +1. This value can appear alternatively as a *threshold* θ_k in the activation level. In this case, u_o is constant and equal to -1 while the weight w_{0k} is equal to θ_k.

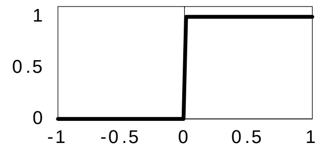
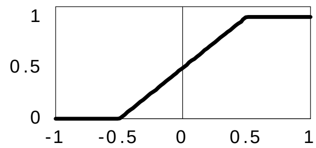
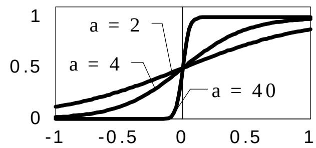
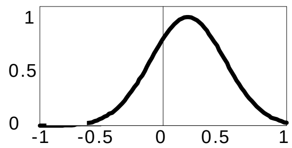
2. the shadowed circle represents the cell body (soma) of the neuron u_k. The electrical reactions in a natural neuron are emulated by a *linear combiner* or *adder*, the sum of the inputs weighted by the strength of the correspondent connections.

3. the activation level of a neuron is determined by an *activation function* $\varphi(.)$. This function is also called *squashing function* because it limits the amplitude of the neuron output into an interval (usually either [-1,1] or [0,1]).

In Table 3-2 we present four basic forms of activation function. The *threshold* function models the "all-or-nothing" activation property described by the seminal work of Mculloch and Pitts [MCCU43]. Usually, the *linear* function has an amplitude of 1 which, when made infinitely large, leads the linear to the threshold function. The *sigmoid* function might assume different forms and it is the most used in neural net algorithms. Sigmoid yields a continuous output and it is differentiable (an important feature in neural networks models). The *gaussian* function is used by the Radial Basis Function algorithm and it is the only one that does not calculate the weighted sum v_k. Rather than weights, the connections describe *j* coordinates of a vector c_k

(called *center*). The norm E_k of the vector (u_j - c_k) and its standard deviation σ are the independent variables of the gaussian function and its result is the activation level of the neuron u_k.

Table 3-2: Types of Activation Function.

Function	Equation	Graphic
threshold	$f(v_k) = \begin{cases} 1, \text{ if } v_k \geq 0 \\ 0, \text{ if } v_k < 0 \end{cases}$	
linear	$f(v_k) = \begin{cases} 1, \text{ if } v_k \geq 0.5 \\ v_k, \text{ if } -0.5 \geq v_k > 0.5 \\ 0, \text{ if } v_k \leq -0.5 \end{cases}$	
sigmoid	$f(v_k) = \dfrac{1}{1 + \exp(-a \cdot v_k)}$	
gaussian	$f(E_k) = \exp\left(-\left(E_k/2\sigma\right)^2\right)$, where $E_k = \left(\sum_j (y_j - c_{jk})^2\right)^{1/2} \geq 0, \ \sigma > 0$	

3.4.2. Neural Network Architectures

Artificial neural cells are organized as graphs whose structure defines the neural network architecture. There are four types of network architectures: *single-layer feedforward* networks, *multilayer feedforward* networks, *recurrent* networks, and *lattice structures*. The modules developed in this dissertation are multilayer feedforward networks. Feedforward networks are processed from the input to the output layer (Figure). The outputs signals of a layer are used as inputs by the adjacent layer. Neural networks cane be either *fully connected* (the cells in all layers are connected to all cells in the adjacent layer) or *partially connected* (otherwise).

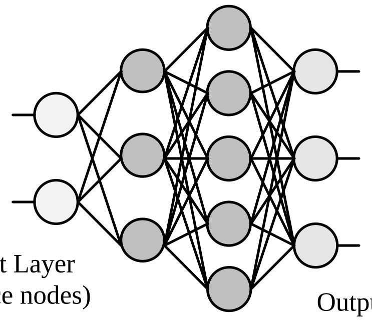


Figure 3-3: Feedforward Multilayer Network

The “knowledge” of a neural network is stored at the connection level in the weight vectors. When trained according to *supervised learning*, a neural network will change its connections following some *error-correction* rule, that is an algorithm that searches for minimizing the network error (difference between target and network response).

The algorithms studied and applied in this dissertation are supervised methods. Both *Radial Basis Function* and *Backpropagation* have algorithms derived from the *error-correction* rule and are trained in a *supervised* manner. On the other hand, while the Backpropagation architectures are determined before the training (fixed-network), the number of hidden neurons in Radial Basis Function can be a step of the learning process (free-network).

3.4.3. Neural Network Algorithms

It became almost impossible to identify the entire list of neural net algorithms available. Constantly, publications have either suggested modifications in known models or presented alternative algorithms. In this work, only the two neural network models developed are discussed.

a) Roots of Error-correction Learning

A neuron network might be described mathematically as a mapping between two spaces. Given the input space $\mathbf{X}_m$, in the iteration p (p-th pattern), a neural network maps every vector $\mathbf{x}(p)$ onto a correspondent vector $\mathbf{y}(p)$ in the output space $\mathbf{Y}_n$. In supervised learning, there is the additional information that the mapping $\mathbf{y}(p)$ has to be as close as possible to the desired response (target) $\mathbf{t}(p)$. Then, for each input vector $\mathbf{x}(p)$ the error $\mathbf{e}(p)$ is given by the difference: $\mathbf{e}(p) = \mathbf{t}(p) - \mathbf{y}(p)$. This difference forms a vectorial function of the error and describes the error vector for the pattern p, given the target vector $\mathbf{t}(p)$ and the output vector $\mathbf{y}(p)$. Rather than correcting the error at the layer level, however, the learning algorithms are neuron-oriented models. Given an output neuron *k*, the error $\mathbf{e}_k(p)$ in this neuron is given by:

$$\mathbf{e}_k(p) = \mathbf{t}_k(p) - \mathbf{y}_k(p) \qquad (3\text{-}8)$$

The global minimum of Eq. (3-8) is the point where the number of errors in the classifications of the training set is the smallest as possible. The network learning consists in the procedure of adjusting the network reaching this minimum. This adjustment is the correction in the weight vector $\mathbf{w}_k$ of each output neuron u_k such as Eq. (3-8) is minimized. The strategy consists in establishing a *cost function* based on the errors $\mathbf{e}_k(p)$ ($k = 1$,m) which, when minimized, leads to the minimum of Eq. (3-8).

In 1960, Widrow and Hoff developed a cost function as the approximate solution with instantaneous values of the squared errors [WIDR60]:

$$\mathrm{E}(p) = \frac{1}{2}\sum_k e_k{}^2(p) \qquad (3\text{-}9)$$

where *k* is the index of the output neuron u_k and *p* is the *p*-th pattern presented to the net. The error function in the weight space is then a hyperparaboloid. The optimization consists in adjusting the weight vector $\mathbf{w}_k$ towards the minimum error direction, that is $\Delta\mathbf{w}_k$ is in the opposite orientation to the gradient vector of the cost function in Eq. (3-9):

$$\Delta\mathbf{w}_{jk}(p) = -\eta\,\frac{\partial \mathrm{E}(p)}{\partial w_{ik}(p)} \qquad \mathrm{j} = 1,\ldots,\mathrm{m} \text{ and } \mathrm{k} = 1,\ldots,\mathrm{n} \qquad (3\text{-}10)$$

where:

a) *j* is the j-th connection of u_k and η is called *learning-rate parameter* and is responsible for establishing the width of a step. This is essentially a problem-dependent parameter. A learning rate excessively large can lead the algorithm to the instability and far from the convergence while a too small η will drastically increase the convergence time; and

b) the derivative in (3-10) is given by the following product:

$$\frac{\partial \mathrm{E}(p)}{\partial w_{ik}(p)} \; -e_k(p).\varphi_k'(v_k(p)).y_i(p) \qquad (3\text{-}11)$$

where $e_k(p)$ is the error associated to the *p*-th pattern given by Eq. (3-8); $v_k(p)$ is the weighted sum of the input signals in the output cell u_k; $\varphi_k'(v_k(p))$ is the derivative of the activation function $\varphi_k(.)$, and $y_i(n)$ is the output of the neuron u_i in the input layer. When the activation function $\varphi_k(.)$ is linear, the adjustment $\Delta\mathbf{w}_{jk}(p)$ is given by:

$$\Delta\mathbf{w}_{jk}(p) = \eta.e_k(p).\,y_i(p) \qquad (3\text{-}12)$$

which is the Perceptron rule to adapt the weight vector [ROSE58]. Eq. (3-11) is more general and includes the *local gradient*, defined as:

$$\delta_k(p) = e_k(p).\varphi_k'(v_k(p)) \qquad (3\text{-}13)$$

Then, the learning rule can be stated by:

$$\Delta \mathbf{w}_{ik}(p) = \eta.\delta_k(p).y_i(p) \qquad (3\text{-}14)$$

Widrow and Hoff applied their Least-mean-square algorithm on a net with an input and an output layer (ADALINE) [WIDR60]. The error correction could not be applied to hidden layers since the target is not known at this level. Nevertheless, the generalization of Widrow-Hoff rule led to the development of the multilayer Perceptron algorithm called *Backpropagation*.

b) Backpropagation

The delta rule in Eq. (3-10) can be applied only if the correct target is previously known. For this reason, the delta rule was originally applied only to double-layer nets, since, in the hidden layers, the neurons do not have a target. The absence of hidden layers implies in the inability of learning to map simple input-output relationships (such as the XOR problem or nonlinear mappings). Although the necessity of hidden layers was acknowledged for a long time (*e.g.,* [MINS69]), the problem was to develop a powerful learning rule to the hidden units. Backpropagation was the first model to accomplish such difficult task. First described by Paul Werbos in 1974 ([WERB74] and [WERB94]), the model was later developed by Rumelhart, Hinton and Williams [RUME86b]. Neural networks that use the Backpropagation algorithm are also called *Multi-layer perceptrons* (MLP) which means a generalization of the Perceptron network [HAYK94].

The main concept behind the *Backpropagation* algorithm is that the error calculated in the output layer can be “back-propagated” to the net so that the internal layers can use this information to correct their weights. The adjustments follow the *delta rule* based on the backpropagated error and on the contribution that each internal cell does to the final error of the network. The implementation of the Backpropagation algorithm involves a forward pass to estimate the current error and a backward step to adjust the weights.

As usual, there is no processing in the input layer. In the output layer, the processing is identical to the optimization process of the least-mean-square error algorithm, described before. The first step is to calculate the error $e_k(p)$ given by Eq. (3-8). The weights of the last layer are then adjusted according to the delta rule given by Eq. (3-14).

It is in the middle layers that the real power of the Backpropagation model lies. Since the actual targets to hidden cells are unknown, the algorithm attributes "responsibilities" to these units regarding the error in the output layer. This is done by "backpropagating" the error in the output layer and using this value in the calculus of the local gradient $\delta_j(p)$ for each hidden cell u_j, given the *p*-th pattern. To derive the local gradient $\delta_j(p)$ for a hidden cell *j*, one has to start from the following definition (which can be derived from Eq. (3-13)):

$$\delta_j(p) = - \frac{\partial \mathrm{E}(p)}{\partial v_i(p)} \tag{3-15}$$

where the error $\mathrm{E}(p)$ is the sum of the errors $(e_k(p))^2$ at the output layer. The local gradient in Eq. (3-15) is calculated by the chain rule of derivatives and the result is:

$$\delta_j(p) = \varphi_j'(v_j(p)).\left(\sum_{k=1,m} \delta_k(p).w_{kj} \right) \tag{3-16}$$

Eq. (3-16) defines the local gradient to be applied in each hidden cell u_j in order to update the weight vector in the middle layers. This equation reflects the fact the hidden cells u_j are "sharing responsibility" for the errors in the output layer proportionally to their importance (weight w_{jk}) and signal (activation level v_j) to each output cell u_k.

The convergence time is one of the problems of Eq. (3-16) because the local gradient can lead to slow passes to the minimum. Also, a major problem is finding a local minimum that can be recognized by the local gradient as the optimal solution. The last issue is still unsolved in Backpropagation models, but some methods and network conditions (*e.g.,* [GORI92], [FRAS93]) have improved the chances of skipping local minimum (*e.g.,* momentum) and reduced the training time (*e.g.,* adaptive learning rate). Rumelhart, Hinton, and Williams [RUME86b] suggested the inclusion of a *momentum* to adapt the learning rate after one iteration such as the convergence can be accelerated without affecting the stability of the delta rule. This inclusion changes Eq. (3-14) into the so-called *generalized delta rule*:

$$\Delta \mathbf{w}_{ik}(p) = \alpha.\Delta \mathbf{w}_{ik}(p - 1) + \eta.\delta_k(p).y_i(p) \tag{3-17}$$

where α is called *momentum*. This constant allows the network to take into account not only the local gradient but also the last changes in the error surface. When it is zero, the generalized delta rule becomes Eq. (3-14). For the sake of convergence of the weighted time series $\Delta \mathbf{w}_{ik}$, the values of the momentum must be restricted into the range $0 \le |\alpha| < 1$.

The Backpropagation Network

Figure 3-4 depicts the architecture of the Backpropagation network. The input layer has p neurons given by the dimension of the input space. There is no processing at this level of the network. The inputs signals are passed to the hidden layer composed by M neurons. Each neuron receives a weighted sum of the input signals and a bias signal (which can be seen an “extra” input x_0 with full connection - w_{0i} = 1). This sum is the neuron activation level which is the parameter to the activation function φ_i whose value is the neuron output. The same process occurs at the output layer. This layer then compares the target answers with the networks responses, “backpropagating” the errors to the previous layer which applied the generalized delta rule (*i.e.,* Eq. (3-17)) to update the weights.

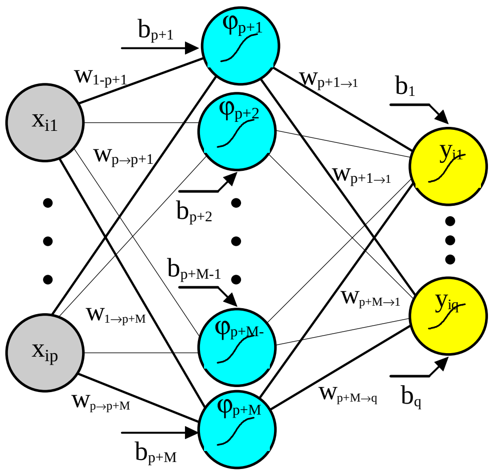


Figure 3-4: Architecture of a Three-layer Backpropagation Networks (MLP).

The Levenberg-Marquardt Method

A different approach is the Levenberg-Marquardt method ([LEVE44] and [MARQ63]) which considers the curvature of the function to establish the step size along the its slope. The strategy consists in expanding the function $f(\mathbf{x},\mathbf{w}) = \mathbf{t}$ into a Taylor series. The purpose is to minimize a cost function $J(\mathbf{w})$ with respect to the weight $\mathbf{w}$. The Newton’s method for this procedure is [HAGA94]:

$$\Delta\mathbf{w} = -\left[\nabla^2 J(\mathbf{w})\right]^{-1} \nabla J(\mathbf{w}) \qquad (3\text{-}18)$$

where $\nabla^2 J(\mathbf{w})$ is the Hessian matrix and $\nabla J(\mathbf{w})$ is the gradient. Particularly, when the cost function is the square sum of the individual errors, that is, when the cost function is given by Eq. (3-8), it can be shown that

$$\nabla J(\mathbf{w}) = \mathbf{J}^{t\cdot}(\mathbf{w}).\mathbf{e}(\mathbf{w}) \tag{3-19}$$

$$\nabla^2 J(\mathbf{w}) = \mathbf{J}^{t\cdot}(\mathbf{w}).\mathbf{J}(\mathrm{w}) + S(\mathbf{w}) \tag{3-20}$$

where $\mathbf{e}$ is the error vector and $\mathbf{J}$ is the Jacobian matrix of derivatives of each error to each weight:

$$\mathbf{J}(\mathbf{w}) = \begin{bmatrix} \frac{\partial e_1(\mathbf{w})}{\partial w_1} & \cdots & \frac{\partial e_1(\mathbf{w})}{\partial w_n} \\ \cdots & \cdots & \cdots \\ \frac{\partial e_N(\mathbf{w})}{\partial w_1} & \cdots & \frac{\partial e_N(\mathbf{w})}{\partial w_n} \end{bmatrix} \tag{3-21}$$

and

$$S(\mathbf{w}) = \sum_{i=1}^{N} e_i(\mathbf{w}) \nabla^2 e_i(\mathbf{w}) \tag{3-22}$$

The Gauss-Newton method assumes that $S(\mathrm{w}) \approx 0$, and the weight updation rule becomes:

$$\Delta\mathbf{w} = (\mathbf{J}^{t\cdot}\mathbf{J})^{-1}\,\mathbf{J}^{t}\mathbf{e} \tag{3-23}$$

The Marquardt-Levenberg modification to the Gaussian-Newton method is:

$$\Delta\mathbf{w} = (\mathbf{J}^{t\cdot}\mathbf{J} + \mu\mathbf{I})^{-1}\,\mathbf{J}^{t}\mathbf{e} \tag{3-24}$$

where μ is a scalar multiplied by some factor (β) whenever a step would result in an increased cost function $J(\mathbf{w})$. A very large μ turns the searching method into the gradient steepest descent with step $1/\mu$. On the other hand, a small factor μ turns the method into Gauss-Newton. The algorithm searches the global minimum by decreasing μ after each iteration. Factor μ is increased only if the error became greater.

The Levenberg-Marquardt Algorithm

The Levenberg-Marquardt algorithm can be described by the following procedures:

1. Initialize the network (*i.e.,* define architecture and random distribution of weights), and present all inputs $\mathbf{x}_i$ and targets $\mathbf{t}_i$;
2. calculate the network response $\mathbf{y}$;
3. compute the sum of the squares of the errors $e_k = \mathbf{y}_k - \mathbf{t}_k$ (cost function $J(\mathbf{w})$).
4. compute the Jacobian matrix.
5. solve Eq. (3-24) to obtain $\Delta\mathbf{w}$.
6. recompute the sum of the squares of errors using $\mathbf{w} + \Delta\mathbf{w}$. If this new sum of squares is smaller than the previous (step iii), reduce factor μ by β, let $\mathbf{w} = \mathbf{w} + \Delta\mathbf{w}$, and return to step I. Otherwise, increase μ by β and return to step v.

7. stop when reach the convergence (the norm of the gradient - Eq. (3-19) - is less than some predetermined value, or when the sum of the squares has been reduced to some error goal).

Although the Levenberg-Marquardt method requires too much memory to adapt the Jacobian matrix, when the network has a few hundred weights, it is very efficient compared to conjugate gradient techniques [HAGA94].

c) Radial Basis Function

Given a set of N points $p_i(\mathbf{x}_i,\mathbf{y}_i)$, Backpropagation searches for a function $f(.)$ such as $f(\mathbf{x}_i) = \mathbf{y}_i$ (for $i = 1$ to N). This characterizes Backpropagation as a *curve-fitting* algorithm based on a given optimization criterion (*e.g.,* gradient descent). Curve-fitting has been a relevant problem for a long time (*e.g.,* [GUES61]). Stated according to *Interpolation* theory, curve-fitting is the following *multivariate interpolation* problem [DAVI63]:

> *Given a set of N different points* $\{\mathbf{x}_i \in \mathbb{R}^p \mid i = 1,...,N\}$ *and a corresponding set of N real points* $\{\mathbf{t}_i \in \mathbb{R}^q \mid i = 1,...,N\}$*, find a function* $F: \mathbb{R}^p \rightarrow \mathbb{R}^q$ *that satisfies the interpolation condition*:

$$F(\mathbf{x}_i) = \mathbf{t}_i,\ i = 1,...N \tag{3-25}$$

There are two forms of interpreting Eq. (3-25): on one hand, $F(\mathbf{x})$ can be conceived as a (linear or non-linear) transformation of the input space $\mathbb{R}^p$ onto the output space $\mathbb{R}^q$; on the other hand, $F(\mathbf{x})$ can be interpreted as the multidimensional surface that divides the hyperspace $\mathbb{R}^p$ x $\mathbb{R}^q$ into regions. While the latter view classifies the search for $F(\mathbf{x})$ as a *Pattern Recognition* problem, the former description is related to *Linear Algebra*. Both theories help to understand the principles of RBF networks. In the next sections, we follow Haykin's description [HAYK94], Poggio and Girossi's [POGG90], and Broomhead and Lowe [BROO88] developments to describe the theoretical framework of radial basis function networks.

The first step to identify the function $F(\mathbf{x})$ that solves Eq. (3-25) was done by Cover in his theorem of *separability of patterns* [COVE65]. This theorem defines a separating surface in the φ space by:

$$\mathbf{w}^T.\varphi(\mathbf{x}) = 0 \tag{3-26}$$

where $$\varphi(\mathbf{x}) = [\varphi_1(\mathbf{x}),...,\varphi_M(\mathbf{x})]^T$$

and $$\mathbf{w} = [w_1,...,w_M]^T$$

In Pattern Recognition, the surface described by Eq. (3-26) is called *decision* (or *descriminant*) *function* since, given a pattern $\mathbf{x}_i$ with unknown classification, one can identify $\mathbf{x}_i$ either as a $\boldsymbol{X}^-$ pattern (if $\mathbf{w}^T.\varphi(\mathbf{x}) \leq 0$) or a $\boldsymbol{X}^+$ pattern (if $\mathbf{w}^T.\varphi(\mathbf{x}) > 0$) [TOU74].

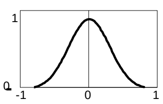
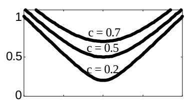
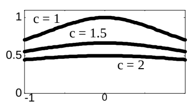
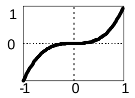
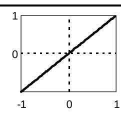
Although Eq. (3-26) does not solve the interpolation problem expressed by Eq. (3-25), it is a rule to the process of classifying the patterns $\mathbf{x}_i$. The patterns can now be separated into an *M*-dimensional space and associated to the output space. This describes the theoretical framework of the first layer of connections in a RBF network. The problem is the following:

- identify the dimension *M* (total of hidden neurons);
- the patterns (*centers*) $\mathbf{c}_j$ (j=1,...,M) that form a basis of an *M*-dimensional space; and
- the function $\varphi(.)$ that describes the separability of the patterns $\mathbf{x}_i$.

Powell proved that a function $\varphi(.)$ can solve Eq. (3-25) as an *exact interpolation problem* (*i.e.*, *M* is equal to *N* in Eq. (3-26)) if its argument is a measure of distance between the centers $\mathbf{c}_j$ and the patterns $\mathbf{x}_i$. In this case, $\varphi(\|.\|)$ is called *radial basis function* [POWE87].

Micchelli [MICC86] proved that, if the radial basis function $\varphi(\|.\|)$ is positive definite and completely monotonic, the Powell's exact interpolation problem has a solution [POGG89]. These findings allow the proposal of a series of radial basis functions to solve the interpolation problem. Table 3-3 there are some examples of radial basis functions that meet Micchelli's theorem.

Table 3-3 - Examples of Radial Basis Functions.

Function	*Gaussian*	*Multiquadratic*	*Inverse Multiquadratic*	*Cubic Splines*	*Linear Splines*
Equation	$\varphi(r) = \exp\left(-r^2/c^2\right)$	$\varphi(r) = \sqrt{r^2 + c^2}$	$\varphi(r) = \left(\sqrt{r^2 + c^2}\right)^{-1}$	$\varphi(r) = r^3$	$\varphi(r) = r$
Graphic					

Powell and Micchelli's findings help to define the kind of function and parameter that can be used to separate the input patterns in the interpolation problem of Eq. (3-25). The next step in the construction of the RBF theory is the definition of the dimension and basis of the M-space of the hyperplane defined by Eq. (3-26). Rather than solve the *exact* interpolation problem, a suitable neural model has to address its *approximate* version. Hence, the interpolation is described as a *hypersurface reconstruction problem* [POGG90], an *ill-posed* problem of learning from a set of training data [HAYK94]. *Regularization theory* [POGG89] transforms this ill-posed to a well-posed problem by minimizing a *cost function* $\mathrm{E}(F)$ involving two terms: one regarding the uniqueness (*generalization error*) and other to the continuity (*regularization error*) of the solution $F(\mathbf{x})$. The unique solution derived by Regularization Theory is the following:

$$\mathbf{w} = (\mathbf{G} + \lambda\mathbf{I})^{-1}.\mathbf{t} \tag{3-27}$$

where λ is the *regularization parameter* that controls the compromise between the degree of smoothness of F and its closeness to the sample data; $\mathbf{t}$ is the target vector; $\mathbf{G}$ is a matrix where a cell $\mathrm{i,j}$ is the Green's function value $G(\mathbf{x}_\mathrm{i},\mathbf{x}_\mathrm{j})$ (i=j=1...N); and $\mathbf{x}_\mathrm{i}$ in $G(\mathbf{x},\mathbf{x}_\mathrm{i})$ is called *center*.

Eq. (3-27) represents the fact that the solution of the regularization problem is in an N-dimensional subspace of smooth functions whose the basis is given by *N* functions $G(\mathbf{x},\mathbf{x}_\mathrm{i})$. In terms of neural network theory, Eq. (3-27) is still not suitable because it solves the exact interpolation problem, that is, the total of centers in the second layer is equal to the number of samples in the training set.

The next step on the derivation of the theory of Radial Basis Functions consists in reducing the dimension of the basis described by the hidden layer of the regularization network. Assuming that the mapping between input and output spaces is not random, there must be a number of samples sufficiently representative to describe this mapping. The dimension reduction is accomplished by approximating the exact solution by the following function:

$$F^*(\mathbf{x}) = \sum_{\mathrm{i}=1}^{M} w_\mathrm{i}.G(\mathbf{x};\mathbf{c}_\mathrm{i}) \qquad \text{or, equivalently: } \mathbf{F}^* = \mathbf{G.w} \tag{3-28}$$

where the vectors $\mathbf{c}_\mathrm{i}$ (*centers*) and the weights $\mathbf{w}_\mathrm{i}$ (i=1,...M) are unknown, the total of centers M is usually lower (and never greater) than the number of samples ($M \leq N$), and the parameter of the Green's function *G* is a norm operator $||\mathbf{x} - \mathbf{c}_\mathrm{i}||$. When the Green's function is a radial basis function, Eq. (3-28) represents the approximate solution to the so-called *Generalized Radial Basis Functions (GRBF) Networks*. The learning rule is derived from minimizing the cost function with respect to the weight vector $\mathbf{w}$, taking the approximate function F^* as the parameter. The solution is given by ([POGG90], [HAYK94]):

$$\mathbf{w} = (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{G}_0)^{-1}.\ \mathbf{G}^T\ \mathbf{t} \qquad (3\text{-}29)$$

where $\mathbf{G}$ is an NxM matrix and $\mathbf{G}_0$ is an MxM matrix where a cell $\mathbf{G}_0$(i,j) is the Green's function value of the distance between the centers $\mathbf{c}_i$ and $\mathbf{c}_j$. When the regularization parameter is zero (*i.e.,* $F(\mathbf{x})$ is assumed to be completely determined by the sample data), the learning rule becomes the approximation method proposed by Broomhead and Lowe [BROO88]:

$$\mathbf{w} = \mathbf{G}^+.\ \mathbf{t} \qquad (3\text{-}30)$$

where $\mathbf{G}^+$ is the *pseudoinverse* of the matrix $\mathbf{G}$ given by:

$$\mathbf{G}^+ = (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{G}_0)^{-1}.\ \mathbf{G}^T \qquad (3\text{-}31)$$

At this point, it is important to address the estimation of the parameter λ. As described before, λ represents the regularization parameter, a measure of the degree to which the prior knowledge about the problem (constraints) should dominate the data. This parameter can be either fixed (according to prior knowledge) or determined by simulations. For instance, Lowe adopted the variance of the noise distribution estimated from the data as a measure of the factor λ) [LOWE94]. Recently, Girosi, Jones and Poggio [GIRO95] presented a Bayesian interpretation of GRBF. The authors describe the interpolation problem as a procedure that maximizes the conditional probability $P(f|g)$, where $f(.)$ is the interpolation function and g the training pattern. By analyzing the probabilities related to $P(f|g)$ and assuming that the individual errors in Eq. () are normally distributed, the authors show that the regularization parameter is equal to:

$$\lambda = 2\sigma^2\alpha \qquad (3\text{-}32)$$

where α is a positive real number and σ^2 is the variance of the individual errors. This equation expresses the trade-off between the level of noise and the strength of the priori assumptions about the interpolation solution. It controls the compromise between the degree of smoothness of the solution and its closeness to the data [GIRO95]. Although Eq. (3-32) gives a hint about the estimation of the regularization parameter, its best value remains problem dependent and even this equation might not work if the error function is not normally distributed (as it happened in the RBF network developed in Chapter 4).

Weighted Norm

Contrary to Regularization Networks, the centers in GRBFs are previously unknown and must be calculated as a step of the learning process. Given a set of N training patterns $\mathbf{x}_i \in \mathbb{R}^N$, the centers must be a set of *M* vectors $\mathbf{c}_j \in \mathbb{R}^p$ that reasonably describes the sample patterns. They can be interpreted as "prototypes" of the patterns $\mathbf{x}_i$ $(i=1,...N)$ [WETT92]. Hence, a key issue in the determination of the centers is the similarity criterion adopted, that is, the norm operator $\|.\|$ chosen. Indeed, without a proper choice of the norm operator, the GRBF can lead to poor generalizations [BOTR91]. For instance, the Euclidean distance takes all individual coordinates x_j $(j=1,...,p)$ of an input vector $\mathbf{x}$ with the original units. This can lead to a poor representation of the input space when some features are more important than others or when scaling factors are not properly considered. Poggio and Girossi [POGG90] proposed a *weighted norm* defined by:

$$\|\mathbf{x}\|_{\mathbf{W}}^2 = (\mathbf{Kx})^T.(\mathbf{Kx}) = \mathbf{x}^T\mathbf{K}^T\mathbf{Kx} \tag{3-33}$$

where $\|\mathbf{x}\|_{\mathbf{K}}^2$ stands for the square of weighted norm with respect to the p-by-p norm weight matrix $\mathbf{K}$. The Euclidean norm is a particular case of the weighted norm derived from Eq. (3-33) when the weight matrix is the identity $\mathbf{I}_{pxp}$. Usually the matrix $\mathbf{K}$ is diagonal whose non-null elements $\mathbf{k}_{ii}$ represent the weight of each input coordinate x_i. These relative weights can be derived either from prior information about the problem or by statistical analysis of the input coordinates of the training patterns.

Particularly, if matrix $\mathbf{K}$ is unknown but all coordinates $\mathbf{x}_i$ have the same strength, and φ is the gaussian function, Poggio and Girossi proved that the best choice (regarding the minimization of the cost function) is a diagonal matrix with each element k_{ii} equal to the inverse of the variance σ_i of each component of the multi-dimensional Gaussian [POGG90]. This is called *Equal Variance Distance*, generally used to overcome scaling and dimension problems of each coordinate [HART75].

Specifically in financial problems, the input features have different grades of importance to the outputs. In this cases, the weighted norm seems to be the more appropriated choice (as we will discuss in Chapter 4). The key issue is then how to translate the weights given by the expert into coefficients of the matrix $\mathbf{K}$ without loosing the self-contained property (*i.e.*, keeping the property that, when all weights are the same, the weighted norm is the Euclidean distance). We started by requiring that the weights attributed by the expert have to add up 1:

Given the vector $\mathbf{x} = [x_1,...,x_p]^t \in \mathbb{R}^p$*, each coordinate* x_i *has a weight* w_i *such as:*

$$\sum_{i=1}^{p} w_i = 1 \tag{3-34}$$

In the limit, when all coordinates x_i have equal weight in the distance, the weight values are :

$$\mathbf{w} = [1/p,...,1/p]^t \tag{3-35}$$

Given the coordinate weights (determined according to prior knowledge) and Eq. (3-33), the problem is to establish the coefficients k_{ii} of the matrix $\mathbf{K}$ (assumed here as a diagonal matrix) such as they reflect the strengths represented by $\mathbf{w}$ and, in the limit, turn the matrix $\mathbf{K}$ into the identity $\mathbf{I}_{pxp}$. Furthermore, matrix $\mathbf{K}$ has to keep the self-contained property also when it is applied to a radial basis function, that is, in the limit, the square of matrix $\mathbf{K}$ has to be the identity $\mathbf{I}_{pxp}$. This last requirement leads to the following equation:

$$\sum_{i=1}^{p} k_{ii}^2 = p \tag{3-36}$$

Combining the conditions in Eq. (3-35) and Eq. (3-36) it is possible to relate the coefficients k_{ii} with the weights w_i by:

$$\sum_{i=1}^{p} w_i = \sum_{i=1}^{p} k_{ii}^2 / p = 1$$

hence:

$$k_{ii} = \sqrt{p.w_i} \tag{3-37}$$

In Chapter 4, we discuss how the weights w_i were established for financial ratios describing the input vectors in the network developed for diagnosing financial health problems.

The RBF Network

A Generalized Radial Basis Function Network is a subcase of a regularization network where the mapping $\mathbf{X} \rightarrow \mathbf{Y}$ is approximated by two space transformations. First, a basis of the input space composed by M centers $\mathbf{c}_j$ is identified. Then, for each pattern $\mathbf{x}_i$ presented to the net, a norm $||\mathbf{x}_i - \mathbf{c}_j||$ is passed as a parameter to the radial basis function associated to the center $\mathbf{c}_j$ (*e.g*, a gaussian function with variance σ_j^2). The outputs of the hidden layer form the matrix **G** and the weight vector **w** (connections to the output layer) is then calculated according to Eq. (3-30). In the last step, the output is calculated as a weighted sum of the hidden layer activations. Figure 3-5 is a schematic view of RBF networks.

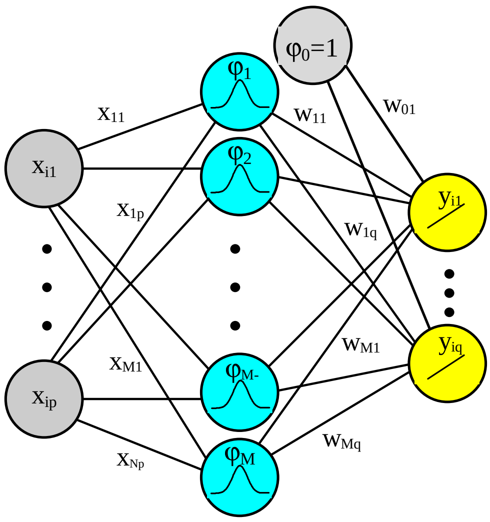


Figure 3-5: Architecture of the Generalized Radial Basis Function (GRBF) Network.

Learning in RBF

From the discussion in the previous sections, one can conceive an RBF network as a special two-layer net that becomes linear in its parameters by fixing all nonlinearities in the first layer of connections. This characterizes the first set of weights as a nonlinear transformation without adjustable parameters and responsible for transforming the input space onto a new space. The transformed space is then linearly combined through adjustable weights [CHEN91]. This identifies two learning processes in an RBF network, one in the first layer of connections and other in the output connections.

Regarding the first connections, it has been showed that the type of the nonlinearity is not as crucial as the chosen centers are [POWE88]. The first phase consists in setting the amount of hidden neurons and the value of their connections to the input layer (*i.e.,* the centers). The aim is to classify the training data into M ($\leq N$) categories (clusters). Both learning strategies can be used: supervised or unsupervised [MOOD89]; and the number of centers is either determined by an algorithm (*e.g.*, Batchelor and Wilkins' method [BATC69]) or it is given as an input data (*e.g.*, the K-means Clustering algorithm [HUSH93]). Several algorithms have been proposed to this task varying from simple random selection of input data to sophisticated pattern recognition strategies.

The first issue is whether apply supervised or unsupervised methods to determine the centers. In one of the first works in this area, Moody and Darken suggested a hybrid learning strategy where the weights in the second layer are determined supervisely (through the gradient descent) and the connections in the first layer (the centers) are self-organized [MOOD89]. The authors proposed the use of the well-known *k-means* clustering algorithm for determining the *k* centers and a *p-nearest* neighbor interpolation method [DASA91] to adjust the widths of these centers. The following criticisms have been made against this approach: (a) the network has a fixed topology which requires a prior determination of the number of centers [BERT95], that is, RBF shares the “number-of-hidden-units” dilemma of multilayer perceptrons [FRIT94]; (b) *p*-nearest neighbor methods neglect the importance of the similarity measure between the patterns; (c) and the method presents a slow run-time and increased memory requirements [LOWE95].

After Moody and Darken’s work, Wettschereck and Dietterich demonstrated the importance of supervised learning to RBF networks [WETT92]. They used Poggio and Girosi’s supervised method [POGG89] to determine the centers according to the gradient descent. In this method, the adjustments in the center positions are established according to the gradient of the error function with respect to the current centers. Although self-organized learning can be useful to skip the local minima problem [LEE91], Wettschereck and Dietterich’s experiments show that the supervised learning of center locations improves the network performance. Nevertheless, when gradient descent is applied to both nonlinear and linear layers, the network can become stuck in local minima [WETT92] and long training times might be necessary [BERT95].

Particularly to this dissertation, the features of the financial diagnostic have indicated the necessity of considering supervised strategies in the process of finding the centers. As we discuss in Chapter 4, the financial diagnostic is not a pure classification problem. The main difficulties are to find a norm that measures the relative weights of each financial variable and to select the best set of firms (centers) to represent the typical problematic situations.

The Orthogonal Least Square Algorithm

A successful supervised learning method is the *Orthogonal Least Square* algorithm, developed by Chen, Cowan and Grant [CHEN91]. Their procedure consists in a supervised selection of centers based on the orthogonal least squares learning (OLS) strategy. The RBF network is viewed as a special case of the linear regression model:

$$\mathbf{t_j} = \sum_{i=1}^{M} \varphi(\mathbf{c}_i - \mathbf{x_j}).\mathbf{w}_i + \varepsilon_j, \text{ for j } = 1,\ldots,N \tag{3-38}$$

where $\mathbf{t}$ is a target vector, $\varphi(.)$ is the radial basis function whose value is known as *regressor*, $\mathbf{c}_i$ is a center, $\mathbf{x}_j$ is an input vector, $\mathbf{w}_i$ is a weight vector and ε_j is the regression error. Chen *et all* treat the problem of selecting centers as an example of how to select a subset of significant regressors from a set of candidates [CHEN91]. Given a set of candidates to be centers, the strategy consists in transforming this set into an orthogonal basis of the hidden layer space. This allows the calculation of the individual contributions that each basis vector does to the desired output energy. We show this procedure by considering Eq. (3-38) in its matrix form:

$$\mathbf{t} = \mathbf{G}.\mathbf{w} + \mathbf{e} \tag{3-39}$$

where

$$\mathbf{t} = [d_1,\ldots,d_N]^t \text{ (N one-dimension output vectors)} \tag{3-40}$$

$$\mathbf{G} = \begin{bmatrix} \varphi_1(\mathbf{x}_1-\mathbf{c}_1) & \ldots & \varphi_M(\mathbf{x}_1-\mathbf{c}_M) \\ \ldots & \ldots & \ldots \\ \varphi_1(\mathbf{x}_N-\mathbf{c}_1) & \ldots & \varphi_M(\mathbf{x}_N-\mathbf{c}_M) \end{bmatrix} \tag{3-41}$$

$$\mathbf{w} = [w_1,\ldots,w_N]^t \tag{3-42}$$

$$\mathbf{e} = [\varepsilon_1,\ldots,\varepsilon_N]^t \tag{3-43}$$

Matrix $\mathbf{G}$ is the regression matrix and can be decomposed in :

$$\mathbf{G} = \mathbf{O.A} \tag{3-44}$$

where $\mathbf{A}$ is an MxM triangular matrix with 1's in the diagonal and the basis vector coefficients α_{iq} (i=1,..,M-1, and q=2,...,M) above the diagonal; and $\mathbf{O}$ is an NxM matrix composed by orthogonal vectors $\mathbf{o}_j$ (j=1,..,M) of the hidden layer space. Both matrixes $\mathbf{A}$ and $\mathbf{O}$ are obtained from the orthogonalization process. Chen et all suggest the use of the classical or modified Gram-Schmidt to find the orthogonal basis $\mathbf{O}$ and the coefficient matrix $\mathbf{A}$. The Gram-Schmidt orthogonalization process can be described by the following procedures:

$$\left.\begin{array}{l} \text{for } k = 2 \text{ to } M: \\ \qquad \mathbf{o}_1 = \varphi_1, \\ \qquad \alpha_{ik} = \mathbf{o}_i^t \varphi_k \big/ \left(\mathbf{o}_i^t \mathbf{o}_i\right), \; 1 \le i \le k-1 \\ \qquad \mathbf{o}_k = \varphi_k - \sum_{i=1}^{k-1} \alpha_{ik} \mathbf{o}_i \\ \text{endfor.} \end{array}\right] \quad (3\text{-}45)$$

This method had been applied before in the context of neural networks as a form to increase the Backpropagation algorithm [ORFA90]. Here, however, the aim is to measure the contribution that each basis vector does by allowing a subselection of centers in the set of M candidates. This is accomplished by the *orthogonal least square* method. Given a set of M candidates to be regressors (centers), after the orthonormalization, the target in Eq. (3-39) can be expressed by [CHEN91]:

$$\mathbf{t} = \mathbf{O}.\mathbf{g} + \mathbf{e} \tag{3-46}$$

where

$$\mathbf{g} = [g_1,...,g_M] \text{ with } g_i = \mathbf{o}_i^t.\mathbf{t} \big/ \left(\mathbf{o}_i^t.\mathbf{o}_i\right), \; 1 \le i \le M \tag{3-47}$$

Chen et all consider the energy $\mathbf{t}^2$ in Eq. (3-46) :

$$\mathbf{t}^t \mathbf{t} = \sum_{i=1}^{M} g_i^2 \mathbf{o}_i^t \mathbf{o}_i + \mathbf{e}^t \mathbf{e} \tag{3-48}$$

With the energy Eq. (3-48), it is possible to obtain the variance of the desired output by:

$$\mathrm{N}^{-1}.\mathbf{t}^t \mathbf{t} = \mathrm{N}^{-1}.\sum_{i=1}^{M} g_i^2 \mathbf{o}_i^t \mathbf{o}_i + \mathrm{N}^{-1}.\mathbf{e}^t \mathbf{e} \tag{3-49}$$

the first term in the left hand of Eq. (3-49) is the contribution of the regressors to the variance of the desired outputs. The other part can not be explained by the regressors. In each iteration, the procedure consists in selecting the regressor with the largest possible contribution to the energy $\mathbf{t}^2$, stopping when the error energy has been reduced to the specified tolerance [SHER92]. Chen et all established the following error reduction ratio due to the regressor $\mathbf{o}_i$ [CHEN91]:

$$[\text{error}]_i = g_i^2 \mathbf{o}_i^t \mathbf{o}_i \big/ \left(\mathbf{t}^t \mathbf{t}\right), \; 1 \le i \le M \tag{3-50}$$

the search for an orthogonal basis is now based on the choice of the most significant regressors, that is, for each iteration i, among the N-i+1 remaining vectors $\mathbf{o}_i$ it is chosen the one that maximizes Eq. (3-50). The algorithm is the following [CHEN91]:

At the first step, for 1 ≤ i ≤ M, compute:

$\mathbf{o}_1^i = \varphi_i$, /* the basis begins with the M RBF values (non-orthogonal) */

$g_1^i = \left(\mathbf{o}_1^i\right)^t \mathbf{t} \Big/ \left(\left(\mathbf{o}_1^i\right)^t \mathbf{o}_1^i\right)$

$[\text{error}]_1^i = \left(g_1^i\right)^2 \left(\mathbf{o}_{1i}^i\right)^t \mathbf{o}_1^{\mathbf{i}} \Big/ \left(\mathbf{t}^t \mathbf{t}\right)$ /*orthogonalization error of each regressor */

Find $[\text{error}]_1^{i1} = \left\{[\text{error}]_1^i,\ 1 \le i \le M\right\}$ /*find the most significant regressor */

and Select $\mathbf{o}_1 = \mathbf{o}_1^{i1} = \varphi_{i1}$, /*select the first orthogonal basis vector */

At the k-th step, with k ≥ 2, for 1 ≤ i ≤ M, and i ≠ $i_1,..i_{k-1}$ (i.e., exclude the previous selections), compute:

$\alpha_{jk}^i = \mathbf{o}_j^t \varphi_i \Big/ \left(\mathbf{o}_j^t \mathbf{o}_{\mathbf{j}}\right),\ 1 \le j \le k-1$ /*orthogonalization coefficient of regressors*/

$\mathbf{o}_k^{\mathbf{i}} = \varphi_i - \sum_{i=1}^{k-1} \alpha_{jk}^i \mathbf{o}_j$ /* Gram-Schmidt orthogonalization process */

$g_k^i = \left(\mathbf{o}_k^i\right)^t \mathbf{t} \Big/ \left(\left(\mathbf{o}_k^i\right)^t \mathbf{o}_k^i\right)$

$[\text{error}]_k^i = \left(g_k^i\right)^2 \left(\mathbf{o}_{ki}^i\right)^t \mathbf{o}_k^{\mathbf{i}} \Big/ \left(\mathbf{t}^t \mathbf{t}\right)$ /*orthogonalization error of each regressor */

Find /* find current regressor */

$[\text{error}]_k^{ik} = \left\{[\text{error}]_k^i,\ 1 \le i \le M,\ i \ne i_1, \ldots, i \ne i_{k-1}\right\}$

and Select $\mathbf{o}_k = \mathbf{o}_k^{\mathbf{i}k} = \varphi_{ik} - \sum_{i=1}^{k-1} \alpha_{jk} \mathbf{o}_j$, where $\alpha_{jk} = \alpha_{jk}^{ik}, 1 \le j \le k-1$

Terminate the procedure after Ms regressor selections when $1 - \sum_{j=1}^{Ms} [\text{err}]_j < \rho$,where $0<\rho<1$

At the end, the centers are the *Ms* regressors that achieved the orthogonalization accuracy. In each iteration, the OLS algorithm increases the dimensional of space spanned by the regressors from (k-1) to k by introducing a new basis vector. The principal merit of the OLS algorithm is to minimize the risk of choosing linear combinations of the chosen centers as additional centers.

3.4.4. Backpropagation or Radial Basis Function?

After analyzing the Backpropagation and RBF algorithms, it remains the issue of which should be the method applied to the diagnostic of financial health problems. Both networks have been applied successfully to financial problems and only a direct comparison of results will define the best choice. Nevertheless, it is possible to anticipate advantages and shortcomings of each model by comparing some general features.

Lowe represents the difference of paradigms between the Backpropagation and RBF models by Figure 3-6. While Backpropagation networks exploit the logistic nonlearity to create combinations of hyperplanes to dissect the pattern space, RBF classify the patterns by modeling *clusters* of data directly and oriented to the information distribution [LOWE95b].

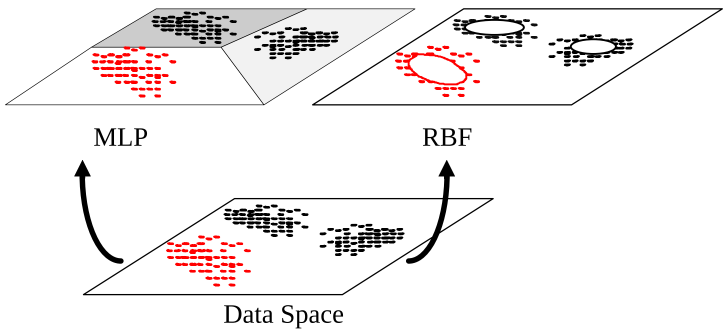


Figure 3-6: Dissection of Pattern Space by Clusters (RBF) and Hyperplanes (MLP). (in [LOWE95])

While Backpropagation may reach local minimum, RBF allows changes on the learning strategies that avoid such situation. On the other hand, MLP performs global approximations of nonlinear mappings $\mathbb{R}^p \to \mathbb{R}^q$ and are able to generalize even with little data. RBF performs on local approximations based on localized nonlinearities. This makes it less sensitive to the order of presentation of samples but it might require a large number of RBF's in some cases. Our strategy will be to implement and test both algorithms and check for the more adequate to the problem.

4

Application:

Financial Health of Small Retail Firms

4.1. Introduction

In this chapter we present the application of this dissertation. First, we present the definition and relevance of the problem in the field of Finance and the choice and generation of the financial variables. The discussion is based on the findings of Martins about the more suitable tools in Finance to diagnose and indicate solutions to financial problems [MART96].

After analyzing the problem according to a financial point of view, we discuss the hybrid neuro-symbolic solution and the proposed architecture to the system [PACH95].

4.2. Financial Statement Analysis

Financial Statement Analysis is an information-processing system designed to provide data to decision makers based on financial statements and on nonaccounting data (*e.g.*, stock prices and aggregate economic indicators) [LEV74]. Its primary and essential function is to convert data into useful information to aid the decision maker in the evaluation of the current and past financial positions and results of the firm's operations [BERN88]. The actual scope of Financial Analysis depends on its purpose, varying from a total analysis of the firm's strengths and weaknesses to a much simpler analysis of its short-term liquidity [BRIG90].

The variety of purposes makes financial analysis rather than a single information system, a flexible method of drawing conclusions about the firm. There are three basic forms of financial reports:

- balance sheet;
- income statement; and
- statement of changes in financial position.

Balance Sheet intends to report the financial position of a firm at a particular point in time by listing its total assets and total liabilities. Therefore, it is a measure based on accounting concepts fixed on a period of time, such as cash, accounts receivable, and inventory. The problem with this accounting approach is that, although the balance sheet may be adequate to measure activities related to short-term financial management, financial decisions require cash flow information rather than accounting numbers. Cash flow is connected with many elements that either do not appear on balance sheet or appear only on its long-term portion [HILL88].

Income Statement reports the profit performance of a firm in a specific period of time in a form of product revenues and factor costs (expenses), that is, the results of operations [WELS77]. The most important measure of the income statement is the *Net Income*. Net Income is the difference between *revenues* (inflows of cash and other elements due to goods sold or services rendered) and *expenses* (outflow of resources, debt payment or taxes related to goods and services used to earn the revenues). In evaluating a firm's performance, the income statement is more important than the balance sheet because it gives greater emphasis upon earnings as compared with asset book values [MCMU79]. Nevertheless, the accrual nature of the accounting elements in the income statement hides the economic profit (*i.e.,* the profit considering the *opportunity cost of capital*) and necessary information to cash flow (*e.g.*, the time when services were rendered or goods were sold). Therefore, several adjustments need to be made in order to reflect both implicit and explicit costs [GALL91].

Statement of Changes in Financial Position is a report of the inflows and outflows of cash. It lists the source of funds (cash inflows) and the use of funds (cash outflows) in a certain period (usually a year). The utility of this statement becomes more clear when it is converted to *Statement of Cash Flows* by incorporating the changes in *net working capital* (current assets-current liabilities). The latter statement is more useful in developing forecast about the amount of cash likely to be spent in the future to satisfy obligations and to evaluate risks. Also, it is restricted to the cash transactions during the period, disregarding other events [GALL91].

The three financial statements represent different and complementary information systems to the financial analysis. Besides the information, the financial analyst has to identify the best framework to the problem. Foster [FOST86] classified the different frameworks according to the objectives of the analysis. A schematic view of Foster's classification is shown in Figure 4-1.

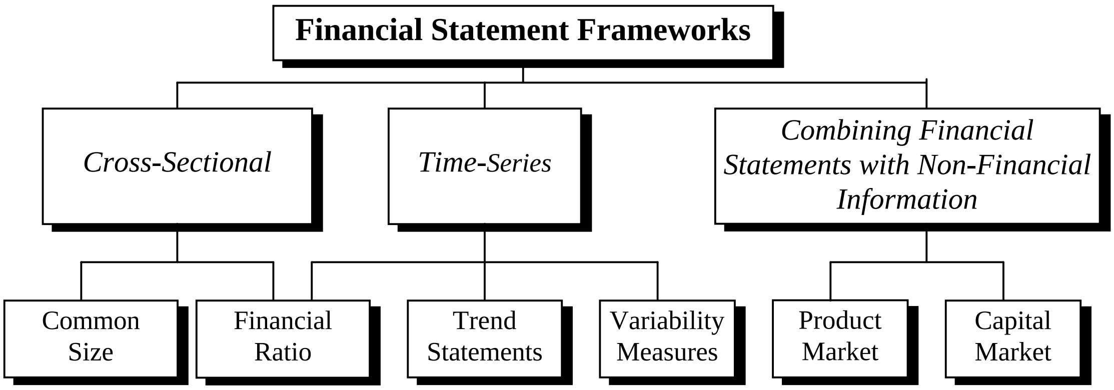


Figure 4-1: Foster's Classification of Financial Statement Analysis Techniques.

The first goal of Financial Statement Analysis is the *Cross-sectional* analysis, that is, the study of firms in a specific economic sector. *Common-Size Statements* and *Financial Ratio Analysis* are two techniques used to compare firms. *Common-Size Statements* tend to reduce the effect of the firm size when comparing financial statements of firms. The strategy is to describe the financial information available (balance sheet, income statement, statement of changes in financial position or other) in percentages. For instance, the balance sheet can be expressed as percentages of the total assets or the income statement as percentages of total revenues [FOST86]. The most widely used method of comparing performances of firms is the *Financial Ratio Analysis*. This technique reduces the amount of information and emphasizes the relationships between financial elements rather than their individual values. Inter-firm differences in financial ratios represent important distinctions in risk and profitability for investors and lenders and an accurate study of ratios can reduce their losses [TAMA78].

Another objective of Financial Statement Analysis is the *Time-Series* analysis. It consists of the study of firm's performances over the time in order to forecast its financial health based on current and past information. Rather than comparing firms' performances, Time-Series uses the firm's information to predict its future developments. Time-Series is accomplished either by *Trend Statement* or by *Financial Ratio Analysis*. Trend Statements are elaborated by fixing a base period and by expressing the financial elements of subsequent periods by their relative values in the base period. A time-series trend can also be identified by studying financial ratios over the time. Basically, these studies isolate individual ratios or combinations of ratios which are observed in search for trends that may forecast failure [GIBS89]. The third approach for time-series is the *variability* analysis where ratio and other variables are also measured over the time. However, in this approach the aim is to define relationships between the extreme values (maximum - minimum) and the mean over the period [FOST86].

The third goal of Financial Analysis is to study firms' performances in order to establish strategies of investments in capital market. This involves the combination of financial statements and non-financial information including *product market* and *capital market*. Product market statements provide information about the market share shifts. Capital markets give a broader range of information, including insights about the changes in expectation of profitability trend, and *dividends payout* (Dividends paid/Net Income) [FOST86]. Both information are important especially for investors and shareholders.

Particularly to the financial problem considered here, Financial Statement Analysis has a twofold use:

i) to diagnose possible financial problems of small firms based on their financial ratios (and trends) and on the corresponding average values of the economic sector; and

ii) to support the deductive strategy to check for causes and offer solutions to financial problems.

Therefore, according to the classification illustrated in Figure 4-1, Financial Statement Analysis is used in this work as a cross-sectional and time-serial technique based on financial ratios. In the following sections we address the classification, functional form, distribution and estimation of financial ratios. Then, the expert's theoretical basis [MART96] for justifying the choice and estimation of the ratios used in the system is discussed.

4.2.1. Financial Ratio Analysis

Financial Ratio is a Financial Statement technique based on proportionate relationships between financial elements X and Y described as X/Y *ratios*. The two reasons for using financial ratios are: to control the size effect when financial variables are examined; and (b) to control industry-wide factors [BARN87]. These indexes can turn the comparison between firms of different sizes more effective and determine the financial status of a firm from different financial perspectives.

The choice and use of financial ratios depend on the purpose of the analysis. Financial ratios are used and perceived differently by commercial loan departments, corporate controllers, certified public accounts and chartered financial analysts. For instance, while loan departments look for debt and liquidity measures, financial executives pay more attention to indexes of profitability [GIBS89].

Salmi and Martikainen [SALM94] divided the theoretical and empirical basis of financial ratio analysis into *functional form, distributional characteristics*, *classification* and *internal rate of return estimation* of financial ratios. Based on the study of these areas, Martins established the criteria for the choice of the financial ratios (*i.e.,* input information) and financial problems (output classification) that compose the neural network module [MART96]. We first address the classification of financial problems and financial ratios. Afterwards we address the Martins' studies and findings concerning the inductive financial ratio analysis applied to financial diagnosis of small retail firms.

4.2.2. Financial Problem Classification

The first issue to be addressed by an expert analyzing the financial health of a firm is the choice of an adequate set of ratios that covers its activities. With this issue comes the classification of the financial problems to be diagnosed.

The process of analyzing financial problems based on ratios can be understood by the metaphor shown in Figure 4-2. Financial ratios act as "flashlights" to specific aspects of financial problems. Some ratios are more enlightening then others like flashlights with a bigger light bulb. There are ratios of difficult classification. They can be used as indicators of more than one problem (though they do not give the same level of information in each case). This could be seen as a fixed flashlight whose moves can illuminate different rooms: the illumination would be different but helpful in both cases.

Salmi and Martikainen [SALM94] classified the methodology for classifying financial problems (the rectangles in Figure) into four approaches: *pragmatical-empiric*, *deductive*, *inductive*, and *confirmatory*.

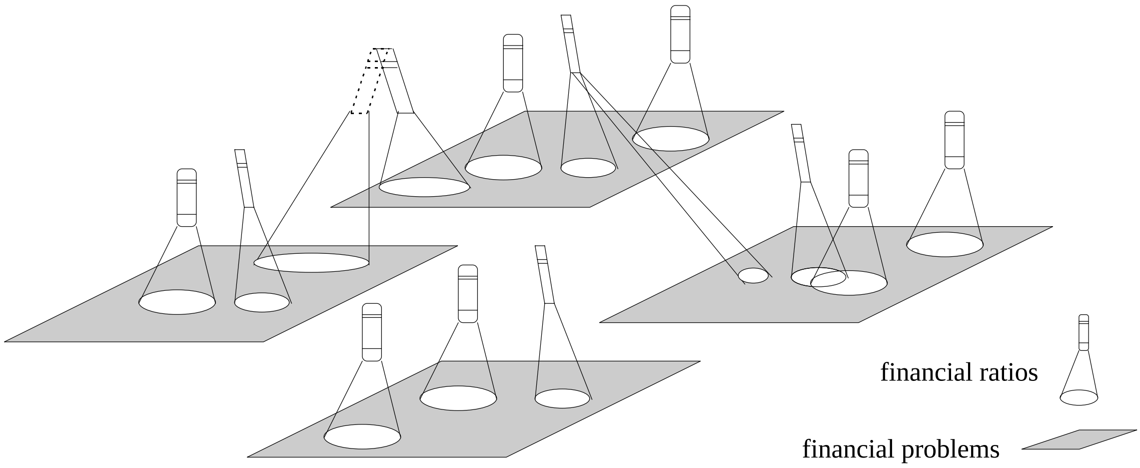


Figure 4-2: Relationship between Financial Ratios and Financial Problems.

The Pragmatical-empiric approach is based on the practical experience of the analyst. There has been no consensus in either the number or the specification of the problem categories. The Deductive approach is based on the *du Pont triangle* system (*i.e.*, the categories profits/total assets, profits/sales, and sales/total assets). Its practical application became limited or mixed with the Confirmatory approach. The Inductive (or data oriented) approach is based on statistical factor analysis. Again, there have been no consensus about the number and the definition of the categories and, particularly in bankruptcy studies, there is instability of financial ratio groups [MARTK94]. The last area identified by Salmi and Martikainen is the Confirmatory approach where a priori ratio classification is hypothesized and checked with empirical evidences built upon statistical tests such as cluster analysis [SALM94].

The main conclusion drawn from Salmi and Martikainen's work [SALM94] is that the best approach for ratio classification is problem-dependent, that is, it depends on the objective of the ratio analysis and on its application environment. Particularly for the purpose of financial health diagnosis of small size firms through inductive reasoning, a suitable approach identified by Martins [MART96] is the Lev's Pragmatic-empirical methodology [LEV74] .

Lev developed a cross-sectional analysis of financial ratios based on four economic aspects of the firm's operations [LEV74]:

- Profitability ratios;
- Short-term solvency (liquidity) ratios;

- Long-term solvency (debt) ratios; and
- Efficiency (turnover or activity) ratios.

Profitability ratios measure the firm's ability to generate earnings [GIBS89]. They were designed to evaluate the firm's efficiency in using stockholders and lenders' capital [LEV74] and in generating its own profits [GALL91]. *Short-term Solvency* or *Liquidity* ratios indicate the firm's ability to meet its short-term obligations [LEV74] or simply the firm's ability to meet cash demands as they appear [GALL91]. The long-run financial and operating structure of a firm is measured by *Long-term solvency* (*Debt* or *Coverage*) ratios. Finally, *Efficiency* (*Turnover* or *Activity*) ratios are indirect measures of cash flow [GALL91].

From the four categories above, three were chosen as representative groups of financial problems that are likely to be warned by financial ratios in the case of small retail businesses. The chosen categories are:

- Profitability;
- Short-term Liquidity; and
- Debt.

Profitability ratios are computed from income statement alone or combined with balance sheet. Liquidity and Debt ratios are computed from balance sheet information [HORN85]. There are two reasons for not assuming Activity problems explicitly: first, when a firm presents activity problems, its operational structure will eventually lead to one or more of the other problems; second, the problem identification through inductive reasoning (neural network module) aims to reduce the space solution in the deductive reasoning (expert system module) rather than presenting the final answer [MART96].

4.2.3. The Financial Ratios Chosen

The number of ratios available to financial analysts increases geometrically with the amount of financial data [HORN85]. Selecting significant ratios is even more difficult than classifying financial problems. The purpose and environment of the financial statement analysis are the keys to chose the ratios. Particularly in financial diagnosis based on inductive analysis, the total of ratios should be large enough to cover the different facets of financial problems and sufficiently small to not overload the amount of information. The criteria for selecting and computing financial ratios are dependent on the kind of business, firm size and homogeneity of the industry sector [MCMU79]. With these restrictions in mind, eight financial ratios were chosen and further studied as measures of financial health of small retail firms [MART96].

In Table 4-1, Table 4-2 and Table 4-3 we describe briefly the financial ratios chosen to measure Profitability, Short-term Liquidity and Debt problems, respectively. These ratios were chosen among tens of financial indicators. Martins justified the choice based on the features of the small retail business. A different kind of firm would certainly require a different set of ratios [MART96]. The chosen indicators offset some shortcomings of classical financial ratios (as the inability of truly measuring resource flows [GALL91]). Horne, Dipchand, and Hanrahan [HORN85] emphasize that no ratio alone can realistically assess the financial condition and performance of a firm, but a group of them can be useful on this task. However, the usefulness of the ratios depends on how the financial analyst perceives their values and relations [PACH95].

Table 4-1: Financial Ratios Related to Profitability Problems.

Ratio	Significance
Cash Flow to Sales	***meaning:*** *Cash Flow = profit before earnings and taxes + depreciation expense + depletion expense + amortization expense. Sales = gross operating receipts - the cost of the returned items.* ***relevance:*** it is a critical indicator of the firm's productivity and creditworthiness, because it measures the relative amount of funds originated by sales and production.
Net Income to Sales	***meaning:*** *Net Income* (profit - taxes and expenses) over *Sales*. ***relevance:*** it is helpful to estimate the Net Income based on sales projection. Particularly in the case of Brazilian retail firms, the flexibility with margin is more restricted. An excessive increase in the margin may lead to loss of market while intensive reductions may cause financial problems [MART96].
Working Capital Turnover	***meaning:*** Sales over average *Working Capital* (current assets - current liabilities). ***relevance:*** It measures how efficiently the working capital is being used to generate revenues. Its main use is to estimate working capital needs given the sales projection [TYRA92].

Table 4-2: Financial Ratios Related to Short-term Liquidity Problems.

Ratio	Significance
Cash Conversion Cycle	***meaning:*** net time interval (in days) between actual cash expenditures on productive resources and the ultimate recovery of cash [PINC90] or, simply: the operating cycle *cash-to-inventory-to-receivable-to-cash.* ***relevance:*** it measures (a) how capable the firm has been in covering the obligations with cash flows from an employment of inventory and receivable; and (b) how sensible is the cash flows to a change in sales or earnings.
T-Working Investment TWI	***meaning:*** Working Capital - Working Investments over Working Investment (which is operating assets - operating liabilities) ***relevance:*** When this ratio is relatively high, the firm has capacity to finance its WI with the resources available in the short term. A relatively low value (as a trend) indicates that the firm is financing its working investment with short term loans.
TWI trend	it confirms or refutes hypothesis regarding TWI depending on the trend. For instance, factors such as seasonally are verified by the trend and not for the single values of TWI.

Table 4-3: Financial Ratios Related to Debt Problems.

Ratio	Significance
Earnings Before Interest and Taxes to Investments	*Earnings Before Interest and Taxes to Interest Expense* (also called *times-interest-earned*) is related to the financial leverage of the firm. The aim is to measure how well the earning capacity of the firm corresponds to its leverage position.
Debt to Equity	The relation between *debt* (from the balance sheet: Liabilities & Equity - Equity) and *equity* (total assets - total liabilities) measures the balance between reduced risk of insolvency by employing mainly equity capital or increased gains (or losses) on equity capital by applying leverage (use of debt) to earn project returns [GALL91].

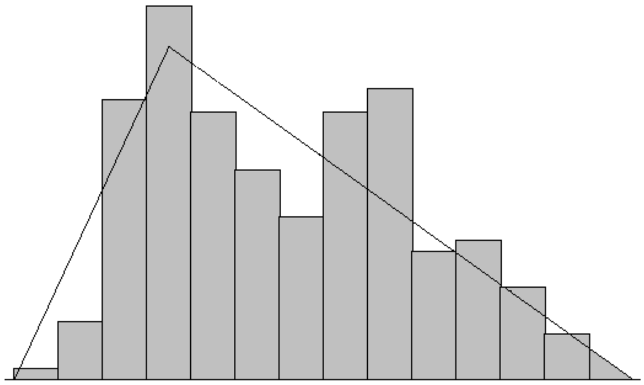
4.2.4. The Random Generation of Samples

Once the ratios have been established, there are two alternatives to the definition of the sample data to train the neural network: (a) collect real data from financial reports of the sample firms; or (b) generate random samples that represent the economic sector to be modeled.

The first alternative is only conceivable when the real data is somehow accessible. Published financial reports (*e.g.*, [COMP95], [SCHO93] and [DUN94]) then become an alternative. Unfortunately, such reports state only the averages of the financial ratios per economic sector and firm size. Although the averages provide the guidance to establish the sector standards, the neural network training has to be based on actual firms' data. The alternative to financial reports is to collect data directly from firms of the economic sector under study. Given a specific economic sector and firm size, the real data available barely surpasses a hundred. Indeed, some studies applying neural networks to bankruptcy prediction have used a few more than a hundred firms as sample data (*e.g.,* [RAGH93], [RAHI93], [ODOM93], [WILS94]). However, the financial health evaluation is a more complex problem. Rather than predicting the death or life of a patient (bankruptcy/not bankruptcy), the aim is to identify the possible illness (profitability, liquidity, debt) for a further treatment. In terms of neural network architecture, this means that the financial diagnosis network requires more output neurons than the single Boolean cell used in bankruptcy models. Consequently, the amount of samples has to be greater as well.

The second alternative to obtain training data is the random generation of samples. The issue is then what probabilistic distributions can represent financial ratios within an economic sector. Although traditionally financial models have been developed under the focus of normality, there are reasons to expect nonnormality for financial ratios [FOST86]. For instance, the distribution can present skewness when the numerator and denominator of the ratio do not hold the hypotheses of strict proportionality or the limits can prevent a normal distribution. Among the several alternatives to solve the problem of nonnormality, one can identify the specific nonnormal distribution by analysis of sample evidence, prior evidence, or by economic analysis of the ratio distribution [FOST86].

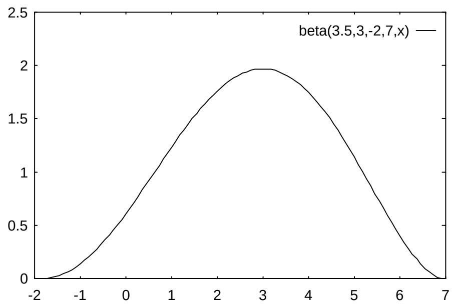
The strategy adopted was to recognize the nonnormality of financial ratios and to study different distributions for the retail grocery sector (SIC 5140). Martins [MART96] studied the distributional forms of financial accounting ratios presented by Buckmaster and Saniga [BUCK90]. The Beta distribution was selected as the initial function for each of the 8 financial ratios. The reason lies on its richness of forms including the J-shape and U-shape functions. Martins studied each ratio and evaluated how this measure can model small firms in the retail grocery sector. Each ratio distribution was established by varying the parameters α and β of the Beta distribution (with mean values taken from the *IRS Corporate Financial Ratios* [SCHO93]). Once the distributions were defined, the samples were generated randomly (according to an algorithm in [DAGP88] pp. 194-195). For each firm generated, a new study was developed comparing its random ratios. The aim was to identify any discrepancy in the relationship between the individual ratios of each firm. Whenever the random ratio values were contradictory, the ratios were changed to keep consistency. Figure 4-3 is a summary of the random generation of the first ratio (see Appendix A for the others). Notice that the modifications in the random values changed the final ratio distribution.



BEST FIT SUMMARY	
Function	**Sq Error**
Triangular	0.0078
Beta (2.1,2.65)	0.0103
Lognormal	0.0112
Erlang	0.0114
Gamma	0.0115
Weibull	0.0122
Normal	0.0165
Uniform	0.0329
Exponential	0.0516

Final Distribution:
mean = 2.53; StdDev = 1.66; min = -0.79; max = 6.37;
histogram range: -1 to 7
Best Fitting: Triangular Distribution: (-1,1,7);
Sq. Error: 0.00783

Original Distribution:
Bucksmaster's U shape
Beta Distribution: beta(3.5, 3);
Chosen interval: [-2,7]



Ratio Limits (Profit and Loss) According to the firm size [SCHO93]:

All		*1M to 999.99M*		*1MM to 24.99MM*		*25MM to 999.99MM*		*100MM and over*	
5.1	2.8	4.0	-0.9	3.4	2.0	5.0	3.1	6.1	4.4

Figure 4-3: Random Distribution of the Financial Ratio *Cash Flow to Sales.*

Three hundred and five (305) retail firms were generated, 200 were used to train (Appendix B) and 105 to test the neural models (Appendix C). In each case, there was the concern of keeping a fair distribution of samples with respect to the financial problems. Table 4-4 is a summary of the distributions of the financial problems.

Table 4-4: Distribution of the Samples Regarding the Financial Problems.

Financial Problem	**Distribution on the Training**		**Distribution on the Testing**	
Profitability	27	13.5%	18	17.2%
Short Term Liquidity	35	17.5%	12	11.4%
Debt	33	16.5%	14	13.3%
Further Analysis	29	14.5%	11	10.5%
Profitability and Short Term Liquidity	21	10.5%	17	16.2%
Profitability and Debt	24	12.0%	19	18.1%
Short Term Liquidity and Debt	31	15.5%	14	13.3%
Total of Firms	**200**	**100%**	**105**	**100%**

4.3. Neural Network: Diagnosing Through Inductive Reasoning

The relation between the indicators and the financial problems depends on the size of the firm, the economic segment, seasonal sales, and business cycle (e.g., [OSTE92], [HAWA86]). Depending on these factors, similar ratios can indicate different financial problems. For instance, the analyst can consider the same ratios with different weights when analyzing distinct economic activities. Besides, as shown in Figure 4-2, although a ratio was related to one category of problem, it can also affect others. For example, when the activity ratio *Net WC/sales* is high it might be an indication of low profitability too.

These are some of the reasons that justify the use of a neural network in the diagnostic phase. It is difficult to establish the direct causal relationship for financial problems. Even if possible the relationship would be too specific and would need to be updated according to the business being analyzed. Updating and learning with indirect relationships are typical advantages of neural networks. Furthermore, the past values of the adjustment ratio influences the future performance of the firm [DAVI93]. Predicting future events based on past data is another task appropriate to neural networks. Particularly in finance, neural networks have performed better than traditional techniques. They have been applied in areas such as bankruptcy forecast [WILS94], prediction of loan chances [BARK90a], and analysis of investments [BARR94].

A close analysis of the financial problem classification allows one to conceive the mapping *ratio-problem*. The problems can be represented by regions on the ratio space. Rather than a single region, a financial problem is represented by areas limited by different relations of ratio values. This means that there are several combinations of ratios that lead to a similar financial problem [MART96]. Considering this fact, it seemed plausible that the Radial-Basis model could represent well this feature by allocating a center to each region of a problem. On the other hand, the limited amount of data was a reason to believe that the Backpropagation model would be more suitable to learn the ratio-problem mapping. Therefore, testing both models became a natural choice.

4.3.1. The Scaling Process

A neural network can learn only if the training data is presented within an appropriate interval. The real data must be scaled into a range that reflects the trends in the original data without affecting the generalization performance. This is accomplished by the scaling process. Specifically in the data sample of the financial ratios collected, the range of each ratio varies according to the data shown Table 4-5. The different nature of the RBF and Backpropagation models required distinct scaling processes. In the next sections we discuss the strategies adopted in each case.

a) Scaling in the RBF network

In the RBF networks, the scaling was established considering three aspects:

- neuron activation;
- critical points and distinguished compressions in the critical intervals; and
- the relative importance of each ratio to each financial problem.

neuron activation. In RBF networks, the neuron activation is a function of the distance among input patterns. Considering the raw input data, the parameter of the radial basis function lies between the interval [minimum distance; maximum distance]. According to Table 4-5 the maximum Euclidean distance between two patterns (firms) is[3]:

$$d_{\max} = \sqrt{\begin{array}{l}\left(6.37-(-0.79)\right)^2+\left(3.21-(-2.81)\right)^2+\left(32.54-(1.91)\right)^2+\left(78.12-(2.26)\right)^2+\\ \left(0.80-(-5.60)\right)^2+\left(1.00-(-1.00)\right)^2+\left(7.82-(5.34)\right)^2+\left(4.98-(-10.11)\right)^2\end{array}}$$

$$d_{\max} \cong 85.$$

there are two problems here: first, this value would not be differentiated by an RBF neuron with a gaussian function. Even the actual maximum Euclidean distances in the training set leads to the following results (taking a fixed variance equal to 1):

for distance = 76.6: $e^{-d^2} = 5.41 \text{ x } 10^{-34}$

for distance = 7.60: $e^{-d^2} = 4.71\text{x}10^{-4}$

---

[3] This theoretical maximum Euclidean distance rarely occurs in practice. The actual maximum Euclidean distance in the training data is 76.6 (90% of the theoretical maximum).

The precision of the calculus has to be sufficiently high to differentiate the activation of the actual maximum distance from the activation of 10% of this maximum. Second, the Euclidean distance is practically established by one ratio alone. The proportional contribution of each ration is: 0.7%, 0.5%, 13%, 79.6%, 0.6%, 0.06%, 2.4%, and 3.2% respectively. This means that ratio 4 is responsible for almost 80% of the maximum distance value. The Euclidean distance has also the disadvantage of mixing units of different coordinates. For instance, while the *Cash Conversion Cycle* is measured in days, the *trend(TWI)* is evaluated by one of the strings [*decreasing, stable, increasing*] represented by the numbers -1, 0 and 1, respectively. Obviously, by mixing such units one yields unreasonable results. A solution for this problem is the adoption of the *Weighted Euclidean Norm*:

Table 4-5: Scaling Situation of Each Financial Ratio.

Ratio	1	2	3	4	5	6	7	8
Name	***Cash Flow/ Sales***	***Net Margin***	***CCC***	***Sales/W***	***TWI***	***trend (TWI)***	***EBIT/I***	***Debt/ Eq***
Problem	Profitab	Profitab	S.T.Liq.	Profitab	S.T.Liq.	S.T.Liq.	Debt	Debt
minimum	-0.79	-2.81	1.91	2.26	-5.60	-1.00	-5.34	-10.11
maximum	6.37	3.21	32.54	78.12	0.80	1.00	7.82	4.98
variance	2.74	1.73	47.65	249.72	2.16	0.39	8.93	8.16
Critical Points	< 2 and > 3.5 30% 30% 2 3.5	< 1.2 > 3.0 30% 1.2 3	< 7.0 > 23 30% 30% 7 23	< 7.0 > 45 30% 30% 7 45	< 0.2 > 0.5 30% 30% 0.2 0.5	-	< 2.0 > 4.5 30% 30% 2 4.5	< 0 > 3 30% 30% 0 3
Scaling Case	II	II	II	II	II	I	II	II

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{p} w_i \times (x_i - y_i)^2} \qquad (4\text{-}1)$$

where **x**, **y** and **w** $\in \mathbb{R}^p$ and **w** is the weight vector that balances the contribution of each coordinate i = 1,...,p in the calculus of the distance. One of the methods to determine the weight vector **w** is to establish w_i as the variance of coordinate i. The resultant distance is then called *Equal Variance Scale*. This distance is invariant with the units of each coordinate and all variables contribute with the same grade to the squared distances [HART75]. The maximum weighted Euclidean distance of the patterns in the training set of the financial ratio problem is given by:

$$d_{\max} = \sqrt{\begin{array}{l}(6.37-(-0.79))^2/2.74+(3.21-(-2.81))^2/1.73+(32.54-(1.91))^2/47.65+\\(78.12-(-2.26))^2/249.72+(0.80-(-5.60))^2/2.16+(1.00-(-1.00))^2/0.39+\\(7.82-(5.34))^2/8.93+(4.98-(-10.11))^2/8.16\end{array}}$$

$$d_{\max} = \sqrt{18.68+20.97+19.69+23.04+19.0+10.24+19.40+27.92}$$

$d_{\max}$ = 12.61.

the values in the square roots represent the contribution of each ratio to the maximum distance, that is: 11.7%, 13.2%, 12.4%, 14.5%, 12%, 6.4%, 12.2%, and 17.6%, respectively. The weighted Euclidean norm brought the maximum difference among contributions from 79.6% to 11.2% of the maximum distance.

The affect of the weighted Euclidean distance on the neuron activation can be noticed by observing its actual value in the training set, that is, 8.86. In this case, the gaussian values (for a fixed variance of 1) would be:

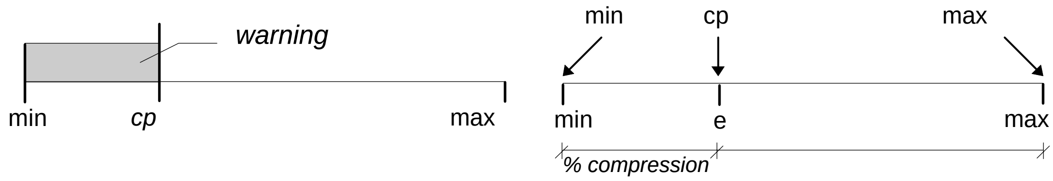
for distance = 8.86: $e^{-d^2} = 1.42 \times 10^{-4}$

for distance = 0.886: $e^{-d^2} = 0.4123$

It can be noticed that 10% of the maximum distance is now much more distinguishable than with the Euclidean norm.

critical points and distinguished compressions in the critical intervals. During the knowledge acquisition process, some critical regions have been established for each financial ratio. A critical region is the vicinity of an imaginary turning point regarding the pattern classification. For instance, in ratio 1, *Cash_Flow/Sales*, the expert identified the neighborhood of 2.0 as the region where a retail company might be close to an insufficient generation of cash. Although this is not a definitive classifier (since other ratios must also be checked before the diagnostic), it is an indication that might be associated to an unsatisfactory profitability.

A ratio has either one (case I) or two critical regions (case II). The choice of the critical point is based on the financial sector and means only a warning that can be even ignored, depending on the other financial ratios. In other words, the critical points can not be seen as financial rules but only as hypothetical landmarks to the classification. The inclusion of critical points aimed to distinguish the scaling factors among ratio intervals. In this way, broader ranges of values with similar patterns can be "compressed", reducing the weighted distance (*i.e.,* approximating similar patterns).



(a) original ratio range. A critical point *cp* is considered the turning point of the ratio interval.

(b) scaling interval. the network input data is preserved into the original interval. The critical point *cp* is scaled into the point *e*. The scaling is differentiated by distinct compression grades in the warning and in the remaining regions.

Figure 4-4: Scaling Case I: (a) original ratio range (b) scaling interval.

According to the critical points, there are two scaling cases. For each, a general scaling equation was established and applied to the ratios. Case I of scaling is illustrated in Figure 4-4a. In this case, the ratio has a single critical point *cp* that corresponds to the turning point *e* in the scaled interval. Once the point *cp* has been established, the formula for scaling the financial ratio values is given by :

$$r_s = \begin{cases} \min + \dfrac{(r - \min)}{(cp - \min)}.(e - \min); & r \le cp \\ e + \dfrac{(r - cp)}{(\max - cp)}.(\max - e); & r > cp \end{cases} \tag{4-2}$$

where $$e = \min + (cp-\min)*(1-comp/100) \tag{4-3}$$

r is the original ratio, r_s its correspondent scaled value, and comp is the percentage of compression of the critical region.

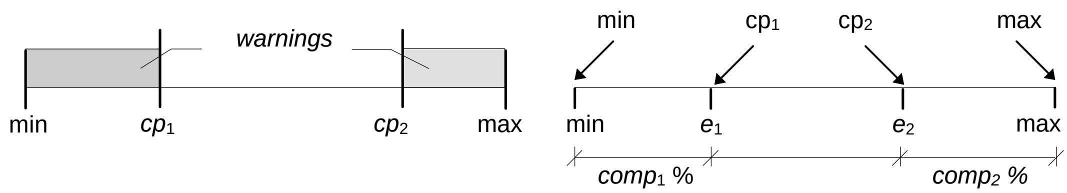
The scaling case II is illustrated in Figure 4-5. Two critical points cp_1 and cp_2 are determined, establishing two "warning regions". Points e_1 and e_2 in the scaling interval are the limits to the compressions of the critical regions 1 and 2, respectively. The scaling of a financial ratio r is given by the following equation:

$$r_s = \begin{cases} \min + \dfrac{(r - \min)}{(cp_1 - \min)}.(e_1 - \min); & r \le cp_1 \\ e_1 + \dfrac{(r - cp_1)}{(cp_2 - cp_1)}.(e_2 - e_1); & cp_1 < r \le cp_2 \\ e_2 + \dfrac{(r - cp_2)}{(\max - cp_2)}.(\max - cp_2); & r > cp_2 \end{cases} \tag{4-4}$$

where *r* is the original ratio, r_s its correspondent scaled value, and $comp_1$ and $comp_2$ are the percentages of compression of the critical regions 1 and 2, respectively. The scaled value e_1 is determined by Eq. (4-3) and e_2 by the following:

$$e_2 = \max - (\max - cp_2)*(1-comp_2/100) \quad (4\text{-}5)$$

In Table 4-6 we present the critical points and grades of reduction adopted in the RBF model. In fact, several grades of reduction were tested and 30% yielded the best results regarding the tradeoff training-testing errors.



(a) original ratio range. Two critical points cp_1 and cp_2 are considered turning points of the ratio interval.

(b) scaling interval. the original interval is preserved. The two critical points cp_1 and cp_2 are scaled into e_1 and e_2, respectively. Ratios in the ranges *min*-e_1 and e_2-*max* are compressed according to $comp_1$ and $comp_2$ grades, respectively.

Figure 4-5: Scaling Case II: (a) original ratio range (b) scaling interval.

relative importance of each ratio. The weighted Euclidean distance reduces the effects of different units and intervals of each coordinate. Each coordinate has the same strength in the final distance. Nevertheless, there still remain an important issue when the weighted Euclidean distance is applied: do all coordinates count with the same grade in the measure of similarity? Particularly in our case, this is not true. As we see in the next section, a single RBF network was unable to learn the patterns of the firm. Three network were designed and trained independently, one for each financial problem (Further Analysis is a result from the activation of the other output neurons). Therefore, there was need for considering each ratio differently in each financial problem. The weight of each ratio depends on the similarity criteria. For instance, if two firms are being compared regarding their profitability condition, ratios 1, 2, and 4 in Table 4-7 should weight more than the others. The strategy here was to identify the weight of each ratio in each financial problem. The individual weights were established with two rules in mind: they should reflect the importance of the ratios and should add up 1. The expert decided, first, to identify clusters of importance (according to the nature of the ratios) and divide the weights within each cluster.

Table 4-6 : Scaling decisions for each financial ratio.

ratio	c_1	$Comp_1$	financial warning	c_2	$Comp_2$	financial warning	e_1	e_2
1	2.0	30%	improper cash generation.	3.5	30%	-	-	-
2	1.2	30%	low profitability.	3	30%	-	-	-
3	7	30%	high turnover (bad for period and sector considered, but it can be favorable in some circumstances)	23	30%	too much capital tied up to the short term cycle.		
4	7	30%	low working capital turnover.	45	30%	high financial cost in order to run the short term activity		
5	0.2	30%	the short term debt is either high or compromises the entire cash.	0.5	30%	-	-	-
6	0	0%	watch 5	0	0	-	-1	1
7	2	30%	earnings insufficient to fulfill obligations.	4.5	30%	-	-	-
8	0	30%	high business risk premium (Equity < 0).	3	30%	high financial risk (premium)	0.5	0.8

The ratio weights in Table 4-7 were determined for each similarity criterion (*i.e.,* for each financial problem) according to the following empirical strategy: first, the ratios were grouped into three or four categories. Each category was weighted based on its affect on the financial problem. The weights within the categories were then established in order to describe their relative importance and to add up the category weight. For each financial problem this process was repeated until the expert was satisfied with the relative importance of each financial ratio. The percentages in Table 4-7 are the parameters w_i in Eq. 3-28 to determine the final weights k_{ii} used in the calculus of the distance[4].

[4] According to EQUATION 3-27, for each financial problem the sum of eight coefficients k_{ii}^2 is equal to 8, the total of financial ratios (the dimension of the input space).

Table 4-7: Ratio Weight Factors for Each Financial Problem

		Financial Ratios							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Financial Problem		***Cash Flow/ Sales***	***Net Margin***	***CCC***	***Sales/ W***	***TWI***	***trend (TWI)***	***EBIT/I***	***Debt/ Eq***
Profitability	w_{ii}	40%	20%	6%	9%	5%	0%	10%	10%
		60%		*15%*		*5%*		*20%*	
	k_{ii}	1.7889	1.2649	0.6928	0.8485	0.6325	0	0.8944	0.8944
Short Term Liq.	w_{ii}	12%	5%	15%	10%	30%	13%	7.5%	7.5%
		17%		*68%*				*15%*	
	k_{ii}	0.9798	0.6325	1.0954	0.8944	1.5492	1.0198	0.7746	0.7746
Debt	w_{ii}	15%	5%	5%	8%	5%	2%	20%	40%
		20%		*20%*				*60%*	
	k_{ii}	1.0954	0.6325	0.6325	0.8000	0.6325	0.4000	1.2649	1.7889
Further Analysis	w_{ii}	17%	10%	14%	14%	10%	5%	15%	15%
		27%		*28%*		*15%*		*30%*	
	k_{ii}	1.1662	0.8944	1.0583	1.0583	0.8944	0.635	1.0954	1.095

Some examples of how the distance between two companies was measured are shown in Table 4-8.

Table 4-8: Example of the Calculus of the Distance Between Firms.

Firm	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	Prof	STL	Debt	F.A
111	1.2	-1.5	3.4	25.6	0.35	0	2.20	1.43	1	0	0	0
146	0.2	-0.1	10.6	70.5	0.78	0	-2.2	3.64	1	0	1	0
170	3.6	1.35	7.8	27.6	0.69	0	3.2	2.42	0	0	0	1
222	0.6	0.13	13.9	26.8	0.47	0	4.25	2.48	1	0	0	0
Distances Between Firms 111 and 146 (average = 2.9852)									2.942	2.778	2.893	3.33
Distances Between Firms 111 and 170 (average = 2.4203)									3.402	1.895	1.958	2.43
Distances Between Firms 111 and 222 (average = 1.7735)									1.954	1.740	1.468	1.93
Distances Between Firms 170 and 222 (average = 3.5260)									4.061	3.006	3.333	3.70

b) Scaling in the Backpropagation network

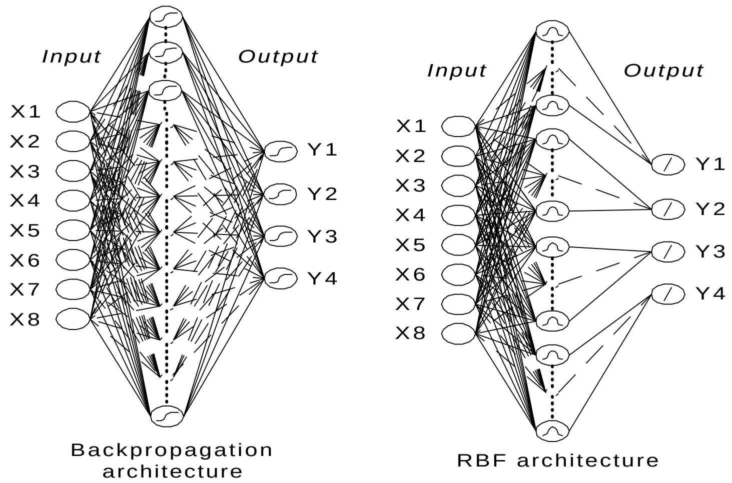


Figure 4-6: The two Neural Network architectures tested.

From the three aspects considered in the scaling in the RBF networks, only one is relevant to the Backpropagation: critical points and distinguished compressions (or expansions) in the critical intervals. The neuron activation is treated directly by the Backpropagation activation function and the parameter is not the distance as it happens in the RBF. Also, since a single Backpropagation architecture was built, all ratios are considered with the same strength.

The first decision regarding the compression (expansion) of ratio intervals was to lead the original maximum and minimum of the training set (*i.e.,* [minT,maxT]) to the scaled limits (*i.e.,* [minS,maxS]) (which is dependent on the activation function). Once the limits were established, one has to recognize the scaling case. As in the RBF model, two cases were identified: (i) intervals with a single critical point; and (ii) intervals with two critical points. In case I, the scaling equation is Eq. (4-2) with the scaling point e given by:

$$e = (maxS\text{-}minS)/2 \qquad (4\text{-}6)$$

In case II, the scaling is given by Eq. (4-4) with the critical points cp_1 and cp_2 and the scaling points e_1 and e_2 determined according to prior knowledge about the ratios.

4.3.2. The Network Architectures

In Figure 4-6 we present the two basic architectures developed. In the Backpropagation model, we varied the scaling intervals, the number of hidden neurons and the hidden and output activation function. The amount of samples combined with the complexity of the diagnose make the convergence an impossible goal to classical Backpropagation models. The only method that did converge was based on the Levenberg-Marquardt algorithm ([LEVE44] and [MARQ63]).

The results of the Backpropagation training are shown in Appendix D. Although several configurations learnt all training samples, their testing performances varied from 38% to 90% of error. The best performances in the testing were 38% and 39% of error with correspondent 7% and 0% of error in the training. The chosen architecture uses the interval [0,1], 10 hidden neurons and logsig as activation function in both layers.

The RBF training required a much broader set of configurations. One can vary the following:

- the learning rule in each layer;
- the grades of compression in the scaling process;
- the method of determining centers;
- the activation function (any kind of RBF function, normalized or not); and,
- particularly in the financial diagnose, the number of networks required for the learning.

The issue of whether apply supervised or unsupervised strategy as the learning rule was solved by testing both approaches. As suggested by the literature (*e.g.,* [WETT92]), the supervised learning was essential to the process of capturing the characteristics of financial situations. The tests confirmed that, without taking into account the financial problem associated with a set of ratios, the network is incapable of identifying problematic financial situations. We adopted the Orthogonal Least Square [CHEN91] in the first layer. The supervised method improved significantly the RBF performance. Only with a large amount of data an unsupervised method could learn the nature of the relationship between ratios and problems. With a supervised strategy the centers are more typical standards to the different financial problematic situations.

Finally, also due to the relative little amount of samples, a single RBF was unable to learn the different situations that led to a financial problem. Table 4-9 is a summary of the RBF simulations (see Appendix E for the entire list of simulations). The three-network approach was better than the single net and the four network models. In the three-network model, the fourth answer is the result of the combination of the other three. From the results shown in Table 4-9 one can conclude that the RBF performance was not as good as expected. The best testing performance was a 60% of correct answers while the correspondent training performance was 23% of errors. The best net used the first 200 samples as possible centers, with an orthogonalization error of 0.05, reaching 74, 78 and 81 for the total of centers of each subnet. It reached 11% of error in the training and 42% in the testing. The experiments show that the inclusion of a greater number of samples improve the RBF performance.

The results from the neural network simulations can be summarized by the following:

Table 4-9: Main Results of the RBF Net Simulations.

Network	Scaling	λ	Orth. Accuracy	Final Centers				Training		Testing	
								RMS	*% Error*	*RMS*	*% Error*
4, 100 KM p=10	30-30%	0	0.20	100	100	100	100	0.2053	33	0.2900	50
4, 100 KM p=5	30-30%	0	0.10	100	100	100	100	0.2007	34	0.2769	47
4, 200 KM p=5	30-30%	0	0.01	141	129	128	175	0.1970	39	0.2892	53
4, 200 FM	60-60%	0	0.10	15	32	42	80	0.2089	36	0.2406	45
3, 100 KM p=10	30-30%	0	0.20	100	100	100		0.1753	30	0.2990	50
3, 100 KM p=5	30-30%	0	0.2	100	100	100		0.1508	23	0.2827	40
3, 100 FM	60-60%	0	0.1	19	33	100		0.1698	29	0.2470	45
3, 200 KM p=20	30-30%	0	0.01	141	129	128		0.1388	25	0.3242	50
3, 200 KM p=10	30-30%	0	0.01	141	129	128		0.1443	28	0.2986	48
3, 200 FM	30-30%	0	0.10	15	32	42		0.1651	26	0.2341	40
3, 200 FM	30-30%	0	0.05	74	78	81		0.1005	11	0.2619	42

a) RBF models based on unsupervised strategies were unsuccessful in finding the centers. The financial comparison between firms requires a specific kind of weighted norm and it is problem identification dependent. In other words, two companies may seem similar regarding ratio values, but the only way to measure the real relationship between the ratios is taking the financial problem into consideration.

b) In both models, there is a remarkable tradeoff between training and generalization error. Again, this seems to be related to the amount of samples.

c) In both models the scaling and range of input values affect the network performance.

Although Backpropagation has been the best model in terms of the tradeoff training/generalization, the RBF networks seem to be more adequate to a future rule extraction procedure. The nature of the financial statement analysis makes the RBF theoretically more adequate to treat the problem, but the practical restriction of the number of samples did not allow to conclude this fact.

Table 4-10: Causes to be checked according to the financial problems.

Kind of the Problem	Some Potential Causes
Activity	purchases, production process, credit granting , sales, credit terms, fixed assets, seasonal sales, accounts receivable turnover, etc.
Debt	maturity, financial risk, lease payments, interest charges, long term investments, working capital, etc.
Profitability	operational costs, pricing, opportunity costs, administrative expenses, market share, etc.
Activity and Debt	purchases, short term debt, fixed asset financing, etc.
Activity and Profitability	days sales outstanding, indirect costs, inventory turnover, etc.
Debt and Profitability	short term debt, gross margin, operating expenses, debt structure, etc.
Activity, Debt and Profitability	current assets turnover, interest charges, gross margin, etc.

4.4. Fuzzy Expert System: Indicating Solutions Through Deductive Reasoning

The main usefulness of the neural module is to reduce the solution space where detailed causes are searched. If the kind of problem is unknown, the amount of questions to the user would be much higher. Since there is a first clue to the problem, the questions can be directed to related causes instead of general investigations.

Table 4-10 presents some examples of factors checked by the analyst in each problematic situation. The solution process starts with checking all potential causes associated to the diagnosis. Each one is compared to standard patterns of the business environment of the firm (economic sector, size, etc.). Whenever a deviation is found, the analyst inspects related factors and recommends appropriate actions.

The modeling of the financial knowledge required indicated two relevant facts about financial statement analysis:

First, solving a financial problem, once its type is known, is basically a task of tracking causes and indicating adjustments. This task is essentially *deductive* in a sense that it is accomplished by comparing facts with associated patterns. The analyst not only recommends but also justifies his/her advice based on the causes identified in the reasoning process. Deductive reasoning and justification are the main reasons for using expert systems in the solution phase. Expert systems have become popular partially for their ability of performing these tasks [HAYE94].

Second, during the process of elucidating the knowledge, the expert uses linguistic terms to describe most of the variables listed in Table 4-10. Terms such as “high reduction of inventory” and “low interest” were constantly used to describe actions to be taken or information about the economy in a certain time. The strategy was then to specify a *fuzzy expert system* [KAND92] to model the process of indicating alternative solutions to the company.

4.4.1. The Knowledge Acquisition and Representation Processes

The knowledge acquisition consisted in identifying the kind of information and understanding the expert has about the problem. Also, we studied which form of representation could model the way he conceives this knowledge and information. Generally, the knowledge was stated as the following: “IF Profitability AND Interest is Medium and Accounts Receivable in Days of Sales is Very High, THEN Reduce Accounts Receivable is Medium”. There are two important aspects in this fact: first, the expert naturally uses *rules* to describe the knowledge; second, among the information he treats, several are *fuzzy* in nature. This motivated the building of a fuzzy-rule based system [KAND96]. The problem consists now in identifying all variables and rules and defining the correspondent fuzzy memberships and crisp values.

The rule base has three kinds of variables: *variables of state, fuzzy variables and limited variables*. Variables of state are crispy and commonly non-numeric. They define particular conditions of a financial and economic aspects. An example is the *Sales forecast trend*. This variable represents a condition (*e.g.*, *increasing*) and it is neither fuzzy nor numeric. Fuzzy variables are used to model the imprecision of the linguistic terms used by the expert to describe the rules. An example is *Inventory* whose state has been described by terms such as *low, high* or *medium*. Finally, limited variables have crispy values and influence the expert’s decision depending on their relationship with crispy boundaries. An example is the antecedent of the rule “If the *Debt/Equity* ratio is *negative,* then the *financial risk* is *high.”* [PACH96].

The fuzzy memberships were determined from the expert’s experience and from the evaluation of the results. The system received a sample case and the answer was analyzed. This process of output analysis was simplified by the use of SIMULINK, where different alternatives of inputs were entered for analyzing the correspondent outputs (see Appendix F). Whenever the expert disagreed with the answer, the consequents of the inference were reevaluated. The general approach was to change the membership functions of the consequents according to the difference between desired and obtained outputs.

One of the decisions we made is the shape of the fuzzy membership functions. We began by using linear functions and checking the difference between their output and the one generated by non-linear functions. The answers varied in an interval between -5% and 5%. Therefore, we decided to keep the simpler inference process of linear functions since the final answers were not significantly different according to the expert. It is relevant to mention that this could be different if the process of suggesting financial solutions to firms is modeled as a dynamic problem. In this case, gaussian functions could be used to model the dynamic features such as different recommendations according to the country economical situation, or according to structural changes in the firm.

In Figure 4-7 we illustrate an example of the fuzzy membership functions used in the system. The main advantage of triangular and trapezoidal functions is their simplicity in representing a fuzzy concept and the facility of parameter changes.

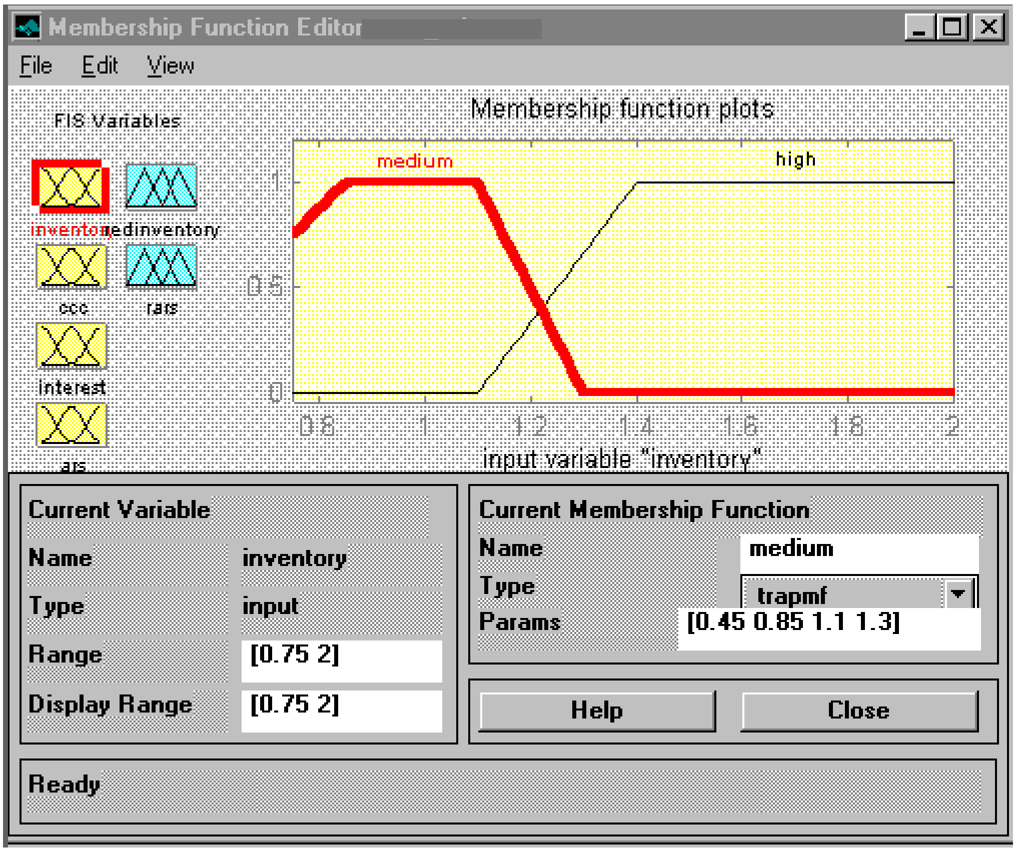


Figure 4-7: Example of the Membership Functions Used in the System.

4.4.2. Approximate Reasoning

The neural network output identifies the kind of rule of the fuzzy expert system. Depending on the financial problem, the consequent of a rule may have different the fuzzy values. The knowledge coming from the neural network is crisp and cannot be evaluated differently by the fuzzy expert system. This limited the fuzzy inference to an Approximate Reasoning deduction.

As in most practical applications of fuzzy systems (*e.g.,* [HIRO93] and [TERA94]), the *max-min* aggregation rule of fuzzy inference [MAMD75] was used to match inputs and fuzzy rules. The fuzzy inference is fired by the input of numeric or state values of the firm's variables. Numeric values will either fulfill numeric variables (*e.g.,* Cash Flow/Sales = 3.1) or give a partial information to a fuzzy variable (*e.g., Inventory* = 1.2). The input can also be the condition of a state variable (*e.g., forecast trend* = stable).

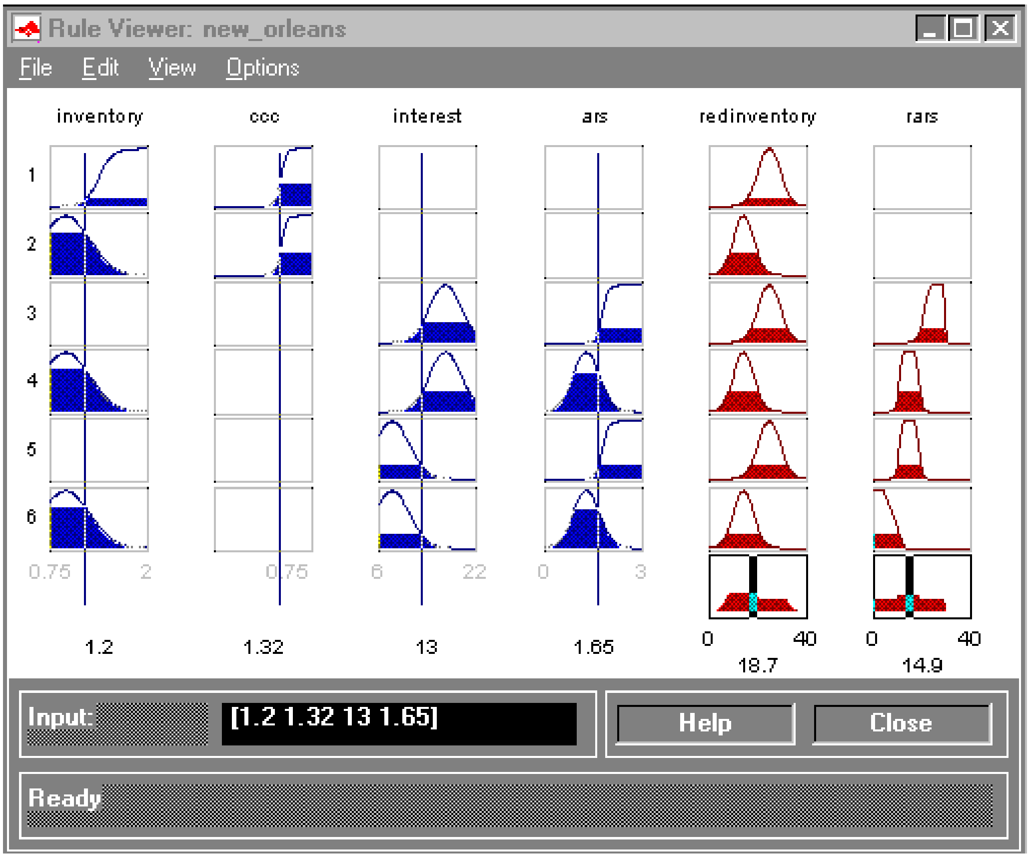


Figure 4-8: Fuzzy Inference Using Nonlinear Sets and MinMax Composition.

Figure 4-8 and Figure 4-9 are comparisons of two different inference process tested in the system: MinMax and ProdMax compositions, respectively. Both were established from nonlinear fuzzy sets. The percentual difference between the two outputs is 2.14% (for the reduction of the inventory) and 0.7% (for the reduction of accounts receivable). As it happen with the fuzzy membership functions, the difference between the results of each fuzzy inference is not significant. For simplicity, we adopted the Min-Max Composition.

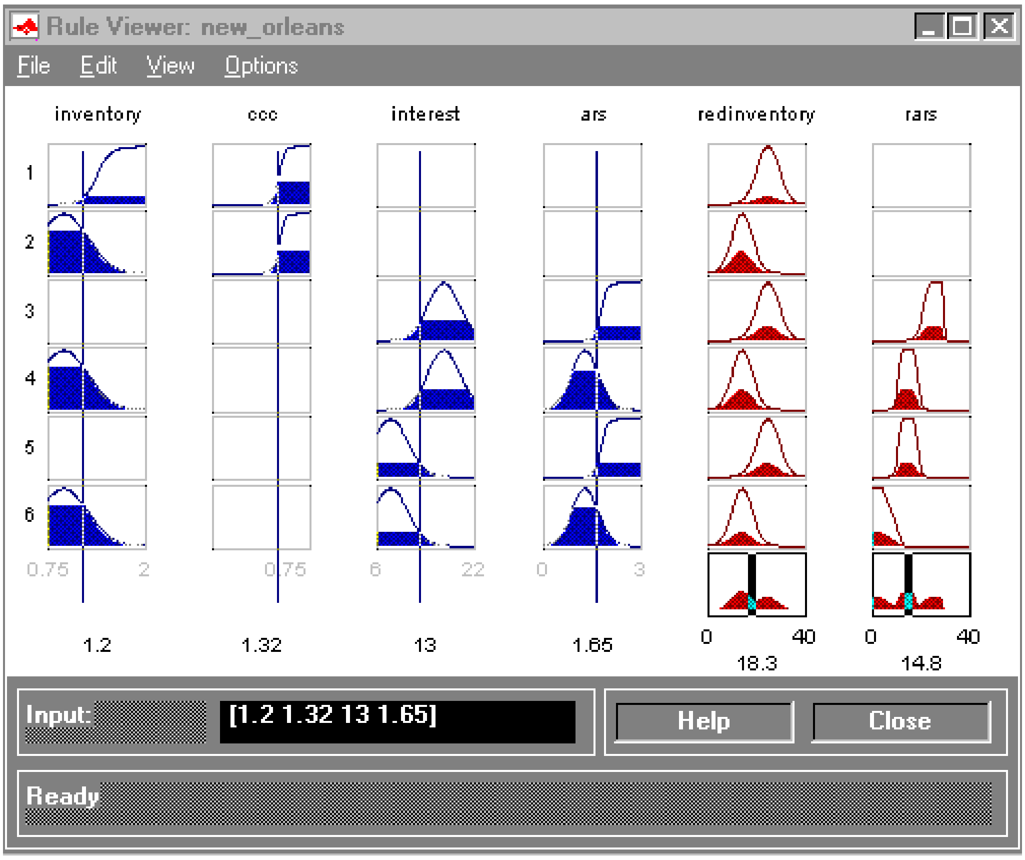


Figure 4-9: Fuzzy Inference Using Nonlinear Sets and ProdMax Composition.

The final answer is the result of the defuzzification of the fuzzy conclusion [PACH96b]. Unfortunately, there is a lack of a systematic approach to the defuzzification process in fuzzy systems [YAGE94], [LEE90]. Several methods have been proposed ([HELL93] and [YAGE93]) varying from the most used *centroid* (*gravity center* or *center of area*) to neuro-defuzzification processes [SONG94]. The centroid was chosen because it keeps the balance between the antecedents, that is, it does not imply different grades of relevance according to the strength of firing, as it happens, for instance, with the “Largest Maximum” and “Smallest Maximum” methods.

The total of rules used in the system was 150. In Appendix G we list 64 of the fuzzy rules used in the system.

4.4.3. The Synergy with Neural Network Module

An issue that can be asked regarding the role of the neural module in the system are: "why the fuzzy expert system needs to use the financial problem as an antecedent of each rule?". The financial problem qualifies the rules. The form of a rule can be determined only if it is specific to the financial problem. As an example consider the rule "IF Liquidity and CCC is High AND Interest is Medium THEN Reduce Inventory is HIGH". If there is no liquidity problem, the consequent could be different (*Medium* Inventory Reduction). Therefore, one of the principal usefulness of the connectionist model is the qualification of the rules of the fuzzy expert system model.

4.5. The Architecture of the Hybrid System

Figure 4-10 depicts the architecture of the hybrid system built in this dissertation. The system is be based on *loosely coupled architecture* [MEDS95] (or *intercommunicating hybrid* system [GOON95b]) for the following reasons:

(a) there is no commercial shell available that includes the kinds of Backpropagation and RBF models needed and a fuzzy expert system module;

(b) the choice for a neural network tool (such as *Matlab*) made possible to test several kinds of neural models;

(c) the development was simplified by the integration based on files; and

(d) different tools were used to find the best models in both parts of the system.

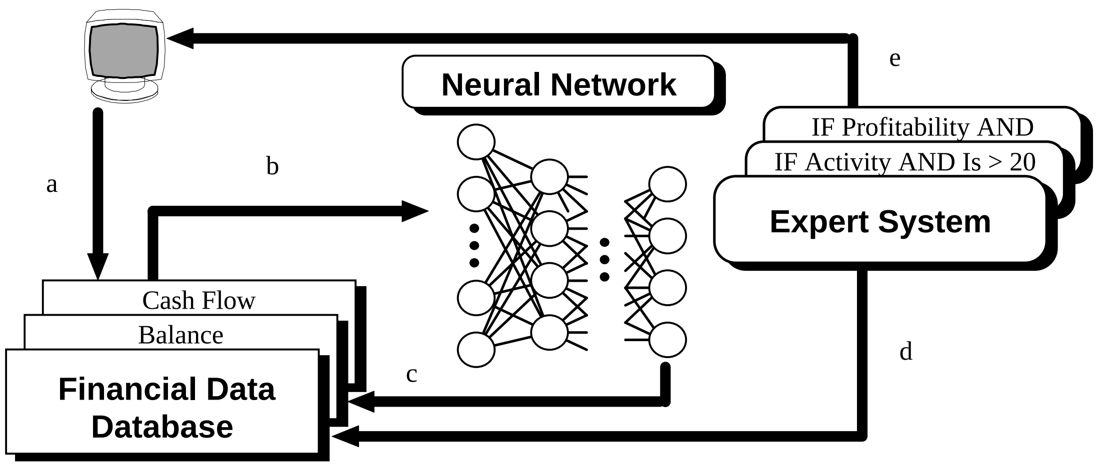


Figure 4-10: The Architecture of the System.
(a) the user enters the data needed to determine the financial ratios**, (b)** the financial ratios are passed to the neural network module which diagnoses the problem, **(c)** this information is sent to the database, **(d)** the expert system consults the database to know the problem and financial indexes**, (e)** after the inference, the suggestions are presented to the user.

The system works in the following way: first, the user informs the financial variables used to calculate the ratios of the firm under consideration; the neural model estimates the financial condition of the firm, indicating a possible problem or a situation where it is not possible to derive a conclusion; the neural output is then saved in file; the fuzzy expert system is now called to proceed an inference in order to indicate a solution to the financial problem.

The financial database is composed by information extracted from the following financial statements

- balance sheet of the company
- cash flow from operations
- income statement

From these statements, the neural module reads the data used to calculate the financial ratios and the fuzzy module reads the data used to calculate reference values in the fuzzy antecedents of the rules. Besides this information, the system asks for trends and forecast to the user.

In Figure 4-11 we present an example of the application of the system, beginning with the neural network output. The inference was applied to a grocery firm whose manager states that *"…there is a fast growing firm in constant* *evolution…".* Besides this auspicious fact, the firm has been *sending* warning signals to the banks. Recently, this firm has experienced an increasing dependence on external funds to support its working capital requirements.

The first step is the calculus of the financial ratios used by the Neural Network (Table 4-12) to indicate an eventual problem. Table 4-11 shows some of the information used by the Fuzzy Expert System module; part of the information corresponds to the manager's perceptions about some trends of the firm. These trends are not evaluated by the system due to the trade-off general purpose-specificity; that is, unless each firm is analyzed by a financial expert, any type of trend analysis would depend on the manager's perceptions [PACH96b].

Table 4-11: Example of some variables used by the FES module.

Variables	***Internal***		***External***	
	values	*manager's perception*	*sector*	*market*
Seasonal:		NO		
Sales forecast trend:		*S or D***		
average Purchases trend*		*S or D*		
Inventory*	25.7		21.4	
Cash Conversion Cycle (CCC)	9.0		6.8	
Interest				13
Accounts Receivable (AR)*	7.6		4.6	

* in days of sales
** Stable or Decreasing

Table 4-12: Inputs of the Neural Network module

Period*	***(1)***	***(2)***	***(3)***	***I *****
CCC (days)	8.2	8.9	9.8	9.0
EBIT / I	5.55	4.92	4.13	4.87
Sales / WC	3.07	3.05	2.97	3.03
Net Income / Sales (%)	1.89	1.75	1.73	1.79
Cash Flow / Sales (%)	4.29	4.07	4.17	4.18
T/ | WI |	0.00	-0.08	-0.19	-0.09
trend T/ | WI |	-1.00	0.00	-1.00	-0.67
Debt / Equity	0.25	0.26	0.30	0.27

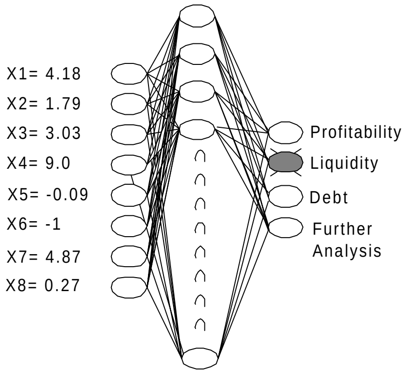
*: bimonthly periods
*Input: average *(no seasonal periods)*

Figure 4-11 is a schematic view of the whole analysis performed by the system. Initially the neural network was called to give a hint about the financial problem. Its output indicated a possible liquidity problem. The fuzzy expert system begun the inference by checking the fuzzy rules whose first antecedent is Liquidity. The inference process led the system to look for (in the database) or ask for (to the user) the information in Table 4-12. The system begins the inference and after composition and defuzzification processes, presents the results. The final response is an advise in order to reduce the firm's average Inventory by 18% and the average

Accounts Receivable by 16%. This means that the firm may slow-down its growth in order to avoid liquidity pressures.

An important aspect of the final diagnostic is its specificity. There is a trade-off between the general purpose of the system and the validation issue. The system reaches its deepest point of analysis when only *rules* particular to the firm could indicate a more detailed diagnostic. For instance, if a firm has low inventory turnover, the system can indicate to “increase the inventory turnover by 8%” but it would not indicate to "increase the inventory turnover by 8% by diminishing the production period in 12%”.

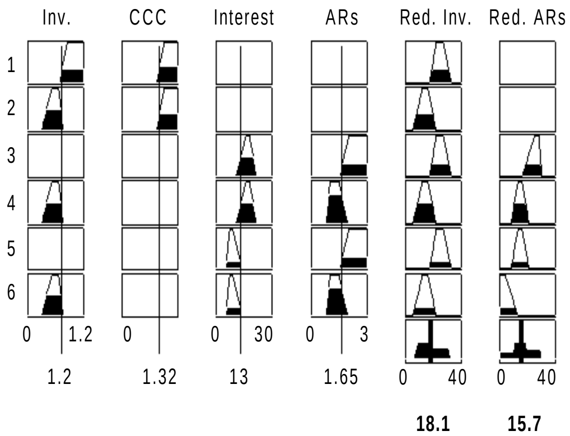
Neural Network Module ***Fuzzy Expert System module***



rules fired

Rules #	Fuzzy Premises (AND) Inv.	CCC	Interest	ARs	Fuzzy Consequence (AND) Reduce Inv.	Reduce ARs
1.	HIGH	HIGH			HIGH	
2.	MEDIUM	HIGH			MEDIUM	
3.			HIGH	HIGH	HIGH	HIGH
4.	MEDIUM		HIGH	MEDIUM	MEDIUM	MEDIUM
5.			MEDIUM	HIGH	HIGH	MEDIUM
6.	MEDIUM		MEDIUM	MEDIUM	MEDIUM	LOW

Inv: Inventory; CCC: Cash Conversion Cycle; ARs: Accounts Receivable in days of Sales.



Variables:

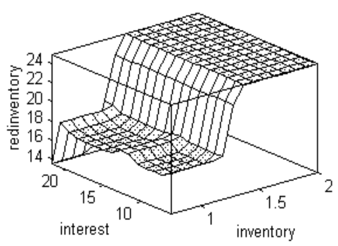
Liquidity, no Seasonal, Sales forecast trend stable or decreasing, Purchases trend stable or increasing
Inventory: I/Io=1.2
Cash Conversion Cycle: CCC / CCCo= 1.32
Interest = 13 % (annual)
Accounts Receivable in days of Sales: ars/arso=1.65

Diagnostic:

1) **Liquidity Problem**
2) **Recommended action(s):**

- **Reduce Inventory by 18 %**
- **Reduce Accounts Receivable by 16 %.**

a)



Liquidity, no Seasonal, Sales forecast trend stable or decreasing, Purchases trend stable or increasing:

diagnostic analysis in relation to the annual interest *(%)* and average inventory *(*measured as *firm’s average inventory / standard average).*

Case a) Inventory reduction
Case b) Accounts Receivable reduction.

b)

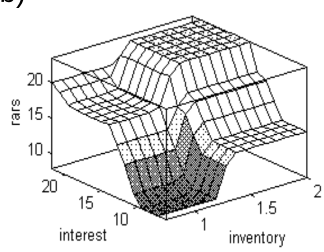


Figure 4-11: Example of Fuzzy Inference in the System.

5

Conclusions and Future Developments

5.1. Conclusions

Our main goal in this dissertation was to develop a system to diagnose and indicate solutions to financial health problems of small and medium firms (SMF). The motivation was the fact that, even though financial management is a critical aspect of surviving for SMFs, they can not afford financial consulting. The development of a computational tool for supporting financial decisions can fulfill this lack. The system can pinpoint the main financial aspects to be adjusted, meaning significant savings of resources to the firm.

The bibliographical search shown that most systems developed in this area were dedicated to bankruptcy and credit analysis. Forecasting bankruptcy and analyzing credit are tasks essentially different than financial health analysis. Bankruptcy models aim to predict the death or life of a firm while credit analysis studies the financial condition of a firm with the exclusive purpose of loan safety. Financial health evaluation is a more complex task: it has to show not whether the patient is about to die but what disease it has; and it is not developed to creditors interested only in the safety of their investments but to managers dedicated to take financial decisions that lead to long-run profitability and success of the firm.

A study of financial statement analysis shown that there are two different reasoning processes participating of the solution: *inductive* and *deductive*. The first clue the expert has about the problem is a consequence of his/her experience in seeing financial scenarios of a firm. The diagnostic is contextual, dynamic with the financial environment and dependent on the expert's experience. It is essentially *intuitive*. On the other hand, the search for causes and correctives follows a logical path of links between financial conditions and actions. It is also dynamic and dependent on the financial environment, but it has the characteristic of being resultant of a *deductive* process.

The choice of the tools that could combine the inductive and deductive reasoning processes lead to the study of *hybrid intelligent systems*, and, more specifically to the development of a *neuro-symbolic* (or *connectionist-symbolic*) system. The inductive reasoning was modeled by the connectionist module while the deduction was implemented through a fuzzy expert system.

The connectionist module was based on financial indicators of the firm. We studied two alternatives of algorithms: Radial Basis Function and Backpropagation. The aim was to find which model would more adequate to combine the heterogeneous features of financial analysis, mainly the dependency of the ratio relevance with the financial problem. Although theoretically we had reasons to believe that the RBF model would be more adequate, the Backpropagation results were better[5]. The main reason is the relative little amount of samples used to train the network. This is actually a practical restriction since a specific economic sector rarely has hundreds of firms of certain size. The alternative is to generate more samples from the study of the financial indicators of the existing firms.

The fuzzy expert system was based on the Approximate Reasoning. Fuzzy and crisp values used by the expert during the deduction are modeled as antecedents of fuzzy rules. Therefore, we used a fuzzy rule-based system based on Approximate Reasoning. The validation was dependent on the trade-off between the general purpose of the system and its level of specificity. The deepest point of analysis is reached when only rules specific to the firm could indicate a more detailed diagnostic.

5.2. Future Developments

5.2.1. Contextual Rules

A significant improvement to the system is the application of some method of rule extraction in the neural phase (*e.g.,* [FU94], [TOWE93], [TOWE94], [ANDR95], [GILE93]). The explicit knowledge of how the network reaches an output would help the financial validation of the results (from a normative point of view) and would give a measure of the degree of structure of the neural knowledge. Also, the translated neural network can be incorporated directly into the fuzzy expert system knowledge base.

---

[5] Percentual errors of each model:

Model	***Training***	***Testing***
Backpropagation	5%	39%
RBF	11%	42%

5.2.2. Fuzzy Neural Network and Possibilistic Reasoning in FES

Another improvement that can be added to the neural module is the transformation of the neural system into a *fuzzy neural network* (*e.g.,* [GUPT92], [JANG95]). The neural output would be not only an indication of a financial problem but also the *possibility degree* of this problem. Given a set of ratios, a fuzzy neural network can form the pairs (problem, possibility degree) for each financial problem. These pairs are read by the fuzzy expert system which the fires the qualitative rules with initial strength given by the possibility degree.

A direct consequence of the development of a fuzzy neural network is that the system can now be modeled as a *possibilistic fuzzy expert system*. The rules modeled currently have all Necessity grade equal to 1, since this was also the supposition to the neural network output. By considering a possibility degree, one can add other rules to the fuzzy expert system, mainly rules non-qualitative, that is, without the financial problem in the antecedents. There are some financial rules whose Necessity degree is not 1, meaning that they occur according to the financial environment. This can be modeled by Possibility theory.

5.2.3. Modeling Market Variables as Fuzzy Intervals

Some variables of the financial statement analysis are being modeled by variables of state (*e.g.,* sales forecast trend = “stable,” “decreasing” or “increasing”). Nevertheless market variables can be modeled by fuzzy intervals rather than by crisp sets. For instance, the knowledge about the interest rate can be described by the interval [0%,10%]. The system can work with different possibilities of interest within this interval. The fuzzy interval can be modeled *flat fuzzy numbers* [DUBO80b] and the Possibility theory can be used to carry the inference.

5.2.4. Integration with Other Modules Developed by the Group

This dissertation is part of a broader work. Since 1991, the group of Applied Artificial Intelligence of the Production Engineering Department of the Federal University of Santa Catarina has been working with Financial Management of small and medium firms. As a result, four master thesis and five doctorate dissertations were developed or are currently being written. Among them, we find the use of Case-Based reasoning for sales forecasting and the application of distributed artificial intelligence for working capital management. These two tasks are directly related to the financial statement analysis and one of the future developments will be the integration between these models.

5.2.5. Development of the System on an Object-Oriented Platform

A recent trend in Hybrid Intelligent Systems has been the integration of different artificial intelligence tools through object-oriented platforms. Several features of the object-oriented paradigm (*e.g.,* the encapsulation, inheritance, polymorphism and reutilization of code) made easier the integration of code and the exchange of information among different technologies [KHEB95]. In order to better fulfill the previous future development, the current system should be rewritten as an object-oriented program in its both modules, the neural and symbolic processes.

5.2.6. Change on the Hybrid System Architecture

Another improvement to the system is the change of its architecture. According to Medsker classification [MEDS94], the system is a *loosely-coupled* system, since it performs the exchange of information between the neural and the symbolic modules via files. The performance will be improved by the implementation of a memory-based information exchange.

Other possibility is the study of an integrated architecture, where the neuron and symbolic modules work as a single processing architecture. According to the Goonatilake and Khebbal's classification [GOON95b], this would mean to transform the system from the current *intercommunicating hybrid* to a *polymorphic hybrid* architecture. The system can work as a *chameleon*, working as a connectionist system during the diagnose phase and as a symbolic process during the solving step.

5.2.7. Use of Different Methods of Neural Learning

The neural module of the hybrid system can be other than RBF or Backpropagation. Actually, even the networks here implemented can be eventually improved by the application of training methods such as *cross-correlation* or data pruning on the sample data. Rather than finding the best neural learning, the aim was to provide means for integrating the neural and fuzzy paradigms in the search for a combined method in financial diagnosis. Future work might improve the learning module or even replace it by extracting the rules from the learnt network (*e.g.*, [ANDR95] and [TOWE93])

6

References

[AMIT89] Amit, D. J., Modeling Brain Function: The World of Attractor Neural Networks, Cambridge: Cambridge University Press, 1989.

[ANDE72] Anderson, J. A., “A Simple Neural Network Generating an Interactive Memory,” *Mathematical Biosciences* 14, pp. 197-220, 1972. (Also in [ANDE89] pp. 181-192).

[ANDE89] Anderson, J. A. and Rosenfeld, E. (editors), Neurocomputing: Foundations of Research, Cambridge: MIT Press, 1989.

[ANDR95] Andrews, R.; Diederich, J.; and Tickle, A. B., “A Survey and Critique of Techniques for Rule Extracting Rules from Trained Artificial Neural Networks,” Neural Computing Research Center, Queensland Univ. of Technology, Australia, *Technical Report* (personal contact to Mr. Diedrich - j.diederich@qut.edu.au), 1995.

[BACH84] Bachant, J. and McDermott, J., “R1 Revisited: Four Years in the Trenches,” *The AI Magazine*, Vol. 5, No. 3, 1984.

[BARC91a] Barcia, Ricardo M.; Luna, Paulo T; Montenegro, Fernando B. & Pacheco, Roberto C. S., “A Working Capital Management Decision Support System - SECAGI,” *TIMS XXXISOBRAPO - Joint Meeting - Rio de Janeiro, July 15-17 - 1991.*

[BARC91b] Barcia, Ricardo M.; Luna, Paulo T; Montenegro, Fernando B. & Pacheco, Roberto C. S., “Treatment of Uncertainty and Subjectivism on a Working Capital Management DSS,” *TIMS XXXISOBRAPO - Joint Meeting - Rio de Janeiro, July 15-17 - 1991.*

[BARC92] Barcia, Ricardo M.; Pacheco, Roberto C. S. & Rodrigues, Alejandro M., “Tratamento de Imprecisão em um Sistema Especialista e Administração de Capital de Giro,” TIMS XXXISOBRAPO - Joint Meeting - Rio de Janeiro, July 15-17 - 1991.

[BARK90a] Barker, D., "Analysing Financial Health: Integrating Neural Networks and Expert Systems," *PC AI,* May/June, 1990.

[BARK90b] Barker, D., "Neural Network and Expert Systems," *AI Week,* Vol. 7, No. 9, 1990.

[BARN87] Barnes, P., “The Analysis and Use of Financial Ratios: A Review Article,” *Journal of Business Finance & Accounting*, Vol. 14, No. 4, pp. 449-461, 1987.

[BARR94] Barr, D. and Mani, G., Using Neural Nets to Manage Investments, *AI Expert*, Vol. 9, No. 2, February, 1994.

[BATA90] Batalha, M. O. and Demori, F., A Pequena e Média Indústria em Santa Catarina, Florianópolis, SC: Editora da Universidade Federal de Santa Catarina (UFSC), 1990.

[BATC69] Batchelor, B. G. and Wilkins, B. R., “Method for Location of Clusters of Patterns to Initialize a Learning Machine,” *Electronic Letter*, Vol. 5, No. 20, pp. 481-483, 1969.

[BELL73] Bellman, R. and Giertz, M., “On the Analytic Formalism of the Theory of Fuzzy Sets,” Information Sciences, Vol. 5, pp. 149-156, 1973.

[BENA90] Benachenhou, D.; Cader, M.; Szu, H.; Medsker, L.; Wittwer, C. & Garling, D., "Neural Networks for Computing Invariant Clustering of a Large Open Set of DNA-PCR Primers Generated by a feature-knowledge based system," in *Proceedings of IJCNNN-90*, San Diego, CA, Vol. II, pp. 83-89, 1990.

[BERN88] Bernstein, L. A., *Financial Statement Analysis: Theory, Application, and Interpretation*, 4th Ed., Homewood, ILL: Irwin, 1988.

[BERT95] Berthold, M. H. and Diamond, J., "Boosting the Performance of RBF Networks with Dynamic Decay Adjustment," *in Advances in Neural Information Processing 7*, Ed. by G. Tesauro, D. Touretzky and J. Alspector, New York : J. Wiley, 1995.

[BEST89] Best, J. H., *Cognitive Psychology*, St. Paul: West Publishing Co., 2nd. ed., 1989.

[BIGU90] Bigus, J. P. & Goolsbey, K., "Integrating Neural Networks and Knowledge-Based Systems in a Commercial Environment," *Proceedings of IJCNN-90,* Washington DC, Vol. II, pp. 463-466, 1990.

[BLOC90] Blocher, E., “ANSWERS: An Expert System for Financial Analysis,” in [LIEB90], pp. 101-125, 1990.

[BONC93] Bonchiev, V.; Vasilev, V.; Popov, I.; Popchev, I.; Simon, G. and Tadzher, V., “Constructing Expert Systems by Using Object-Oriented Programming,” *J. of Computer and Syst. Sciences Int.*, Vol. 31, No. 3, pp. 57-64, 1993.

[BOTR91] Botros, S. M. and Atkeson, C. G., “Generalization Properties of Radial Basis Functions,” In *Advances In Neural Information Processing Systems 3*, ed. by J. E. Moody, R. P. Lippman, and D. S. Touretzky, San Mateo, CA : Morgan Kaufmann, pp. 707-713, 1991.

[BOUC93] Bouchon-Meunier, B., Valverde, L. and Yager, R. R. (editors), *Uncertainty in Intelligent Systems*, Amsterdam: North-Holland, 1993.

[BRIG90] Brigham, E. F.; and Gapenski, L. C., *Intermediate Financial Management*, 3rd Ed., Orlando, FL: The Dryden Press, 1990.

[BROO88] Broomhead, D. S. and Lowe, D., “Multivariable Functional Interpolation and Adaptative Networks,” *Complex Systems*, Vol. 2, pp. 321-355, 1988.

[BUCH78] Buchanan, B. G. an Feigenbaum, E. A., “Dendral and Meta-Dendral: Their Applications Dimensions,” *Artificial Intelligence*, Vol. 11, No. 1, pp. 5-24, 1978.

[BUCH84] Buchanan, B. G. and Shortliffe, E. H. (editors). *Rule-Based Expert Systems : The MYCIN Experiments Of The Stanford Heuristic Programming Project*. Reading, MA: Addison-Wesley, 1984.

[BUCK90] Buckmaster, D. and Saniga, E., “Distributional Forms of Financial Accounting Ratios: Pearson’s and Johnson’s Taxonomies,” *Journal of Economic and Social Measurement*, Vol. 16, pp. 149-166, 1990.

[BULL77] Bullock, T. H.; Orkand, R. and Grinnell, A., *Introduction to Nervous Systems*, San Francisco: Freeman, 1977.

[BURT93] Burton, R. G. (Ed.); *Natural and Artificial Minds*, Albany: SUNY Press, 1993.

[BUTA94] Buta, Paul, “Mining For Financial Knowledge with CBR,” *AI Expert*, vol. 9, No 2, pp.34-41, February - 1994.

[CALL92] Callataÿ, A. M. de; *Natural and Artificial Intelligence: Misconceptions about Brains and Neural Networks*, 2nd. Ed., Amsterdam: North-Holland, 1992.

[CAUD91] Caudill, Maureen, “Expert Networks,” *BYTE,* pp. 108-116, October - 1990.

[CAUD93] Caudill, M., *Neural Networks Primer*, 3rd. Ed., San Francisco, CA: Miller Freeman, 1993.

[CHEN91] Chen, A.; Cowan, F. N.; and Grant, P. M., “Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks,” *IEEE Transactions on Neural Networks*, Vol. 2, No. 2, pp. 302-309, 1991.

[CHER90] Chér, R., A Gerência das Pequenas e Médias Empresas: O que saber para administá-las, São Paulo: editora Maltese, 1990.

[COMP95] *Compustat PC Plus : Corporate Text*, Englewood, Colo.: Standard & Poor's Compustat Services, 1995.

[COVE65] Cover, T. M., “Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications to Pattern Recognition,” *IEEE Transactions On Electronic Computers*, Vol. EC-14, No. 3, pp. 326-334, 1965.

[CZEJ93] Czejdo, Bogdan; Eick, Christoph F. & Taylor, Malcolm, “Integrating Sets, Rules and Data in an Object-Oriented Environment,” *IEEE Expert*, Vol. 8, pp. 59-66, February - 1993.

[DAGP88] Dagpunar, J., *Principles of Random Variate Generation*, Oxford: Claredon Pres, 1988.

[DASA91] Dasarathy, B. V. (editor), *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*, Washington: IEEE Computer Society Press, 1991.

[DAVI63] Davis, P. J., *Interpolation and Approximation*, New York: Blaisdell Pub., 1963.

[DAVI93] Davis, Harry Zvi and Peles, Yoram C., Measuring Equilibrating Forces of Financial Ratios, *The Accounting Review*, Vol. 68, No. 4, October 1993.

[DEMP91] Dempster, M. A. H. & Ireland, A. M., “Object-Oriented Model Integration In A Financial Decision Support System,” *Decision Support Systems,* Vol. 7, No. 4, pp. 329-340, November - 1991.

[DUBO80] Dubois, D. and Prade, H., “New Results About Proprieties and Semantics of Fuzzy Set-Theoretic Operators,” in *Fuzzy Sets - Theory and Applications to Policy Analysis and Information Systems*, edited by P. P. Wang and S. K. Chang, New York: Plenum Press, pp. 59-75, 1980.

[DUBO80b] Dubois, D. and Prade, H., *Fuzzy Sets and Systems: Theory and Applications*, New York: Academic Press, 1980.

[DUBO84] Dubois, D. and Prade, H., “Fuzzy Logics and the Generalized Modus Ponens Revisited,” *Cybernetics and Systems*, Vol. 15, pp. 293-331, 1984.

[DUBO85] Dubois, D. and Prade, H., “A Review of Fuzzy Set Aggregation Connectives,” *Information Sciences*, Vol. 36, pp. 85-121, 1985.

[DUBO88] Dubois, D. and Prade, H.; “An Introduction to Possibilistic and Fuzzy Logics,” In *Non-Standard Logics for Automated Reasoning*, ed. by P. Smets, A. Mamdani, D. Dubois, and H. Prade, San Diego: Academic Press, 1988.

[DUDA76] Duda, R. O.; Hart, P. E. and Nilsson, N. J., “Subjective Bayesian Methods for Rule-Based Inference Systems,” in *Proceedings 1976 National Computer Conference,”* Vol. 45, AFIPS, pp. 1075-1082, 1976.

[DUDA79] Duda, R. O.; Gaschnig, H. and Hart, P., “Model Design in the PROSPECTOR Consultant System for Mineral Exploration,” in *Expert Systems in the Micro-electronic Age,* ed. by D. Michie, Edinburgh University Press, pp. 153-167, 1979.

[DUN94] Dun & Bradstreet, Inc., *Industry Norms and Key Business Ratios - One Year*, Desk-Top Edition, SIC #0100-8999, USA, 1994.

[EMAM91] Emam, K. El; Khalafalla, F. B.; Hoptroff, R. G. & Hall, T. J., “Object Oriented Neural Networks,” *IEE Conference Publication*, Vol. 2, No 332, pub. by IEE, Michael Farady, House, Stevenage, England, pp.1007-1010 - 1991.

[FAHL88] Fahlman, S. E., “An Empirical Study of Learning Speed in Back-Propagation Networks,” Carnigie Mellon University, School of Computer Science, *Technical Report* CMU-CS-88-162, (ftp://128.2.222.79/usro.anon/1988/CMU-CS-88-162.ps), 1988.

[FEIG71] Feigenbaum, E. A., “On the Generality and Problem Solving: A Case Study Using the DENDRAL Program,” In *Machine Intelligence 6*, ed. by B. Meltzer and D. Michie, New York: American Elsevier, pp. 165-190, 1971.

[FEIG92] Feigenbaum, E. A., “A Personal View of Expert Systems: Looking Back and Looking Ahead,” *Knowledge System Laboratory*, Stanford University Report No. KSL 92-41, April 1992.

[FOST86] Foster, G., *Financial Statement Analysis*, 2nd. Ed., Englewood Cliffs NJ: Prentice-Hall, 1986.

[FRAS93] Frasconi, P.; Gori, M.; and Tesi, A., “Successes and Failures of Backpropagation: A Theoretical Investigation,” in *Progress in Neural Networks*, ed. by O. Omidvar, Norwood, N.J. : Ablex Pub., 1993.

[FREE95] Freeman, J. A. S.; and Saad, D., “Learning and Generalisation in Radial Basis Function Networks,” *Neural Computation*, Vol. 7, No. 5, 1995.

[FRIT94] Fritzke, B., “Growing Cell Structures - A Self-organized Network for Unsupervised and Supervised Learning,” *Neural Networks,* Vol. 7, No. 9, pp. 1441-1460, 1994.

[FU94] Fu, L. M., “Rule Generation from Neural Networks,*” IEEE Transactions on Systems, Man and Cybernetics*, Vol. 28, No. 8, pp. 1114-1124, 1994.

[GAIN76] Gaines, B. R., “Foundations of Fuzzy Reasoning,” *International Journal of Man-Machine Studies*, Vol. 8, pp. 623-668, 1976. (Also in [GUPT77], pp. 19-75).

[GALL88] Gallant, Stephen, "Connectionist Expert Systems," *Communications of the ACM*, Vol. 31, No 2, pp. 114-136, February, 1988.

[GALL91] Gallinger, G. W. and Healey, P. B., *Liquidity Analysis and Management*, 2nd. Ed. Reading, MA: Addison-Wesley, 1991.

[GALL93] Gallant, S. I., *Neural Network Learning and Expert Systems*, Cambridge, MA: The MIT Press, 1993.

[GARD93] Gardner, D. (editor), *The Neurobiology of Neural Networks*, Cambridge, MIT Press, 1993.

[GIAR94] Giarratano, J. and Riley, G., *Expert Systems: Principles and Programming*. 2nd. Ed., Boston: PWS Pub., 1994.

[GIBS89] Gibson, C. H., *Financial Statement Analysis: Using Financial Accounting Information*, 4th. Ed., Boston: PWS-KENT Pub. Co., 1989.

[GILE93] Giles, C. L. and Omlin, C. W., “Extraction, Insertion and Refinement of Symbolic Rules in Dynamically Driven Recurrent Neural Networks,” *Connection Science*, Vol. 5, Nos. 3 & 4, pp. 307-337, 1993.

[GIRO95] Girosi, F.; Jones, M.; and Poggio, T.; “Regularization Theory and Neural Networks Architectures,” *Neural Computation*, Vol. 7, pp. 219-269, 1995.

[GOON95] Goonatilake, S. and Khebbal, S. (editors), *Intelligent Hybrid Systems*, New York: John Wiley, 1995.

[GOON95b] Goonatilake, S. and Khebbal, S.,”Intelligent Hybrid Systems: Issues, Classifications, and Future Directions,” in [GOON95], 1995.

[GORI92] Gori, M. and Tesi, A., “On the Problem of Local Minima in Backpropagation,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 1, pp. 76-86, 1992.

[GROS76] Grossberg, S., “Adaptative Pattern Classification and Universal Coding: I. Parallel Development and Coding of Neural Feature Detectors,” *Biological Cybernetics* 23, pp. 121-134. (Also in [ANDE89] pp. 243-258).

[GUES61] Guest, P. G., *Numerical Methods of Curve Fitting,* Cambridge, UK: University Press, 1961.

[GUPT77] Gupta, M.; Saridis, G. N. and Gaines, B. R. (editors), *Fuzzy Automata and Decision Processes*, New York: North-Holland, 1977.

[GUPT85] Gupta, M.; Kandel, A.; Bandler, W. and Kiszka, J. B. (editors), *Approximate Reasoning Expert Systems*, Amsterdam: Elsevier Science Pubs. (North Holland), 1985.

[GUPT92] Gupta, M. M. and Qi, J., “On Fuzzy Neural Models,” in [ZADE92], pp. 479-491, 1992.

[GUTK91] Gutknecht, M.; Pfeifer , R.; and Stolze, M., "Cooperative Hybrid Systems," *Institut fur Informatik Technical Report*, Universitat Zurich, 1991.

[HAGA94] Hagan, M. T.; and Mehnaj, B. M., “Training Feedforward Networks with Marquardt Algorithm,” *IEEE Transactions on Neural Networks*, Vol. 5, No. 6, pp. 989-993, 1994.

[HANS88] Hanson, M. A. & Brekke, R. L., "Workload Management Expert System - Combining Neural Networks and Rule-Based Programming in an Operation Application," *Proceedings of Instruments Society of America,* Vol. 24, pp.1721-1726, 1988.

[HART75] Hartigan, John A., *Clustering Algorithms*, New York: John Wiley & Sons, 1975.

[HAWA86] Hawawini, G.; Viallet, C. and Vora, Ashok, Industry Influence on Corporate Working Capital Decisions, *Sloan Management Review*, Vol. 27, No. 4, Summer, 1986.

[HAYE94] Hayes-Roth, F. and Jacobstein, N., "The State of Knowledge Based Systems," *Communications of ACM*, Vol. 37, No. 3, March, 1994.

[HAYK94] Haykin, S., *Neural Networks: A Comprehensive Foundation*, New York: Macmillan Pub., 1994.

[HEBB49] Hebb, D., *The Organization of Behavior,* New York: Wiley Pub.,1949.

[HECK86] Heckerman, D., “Probabilistic Interpretations for MYCIN’S Certainty Factors,” in [KANA86], pp. 167-196, 1986.

[HECK93] Heck, D., “Rat Cerebellar Cortex in Vitro Responds Specifically to Moving Stimuli,” *Neurosciences Letters*; Vol. 157, No. 1, pp. 95-102, July, 1993.

[HECK95] Heckerman, D.; Mamdani, A. and Wellman, M., “Real-World Applications of Bayesian Networks,” *Communications of the ACM*, Vol. 38, No. 3, pp. 24-26, 1995.

[HELL93] Hellendoorn, H. and Thomas, C., “Defuzzification in Fuzzy Controllers,” *J. of Intelligent & Fuzzy Systems*, Vol. 1, No. 2, pp. 109-123, 1993.

[HEND91] Hendler, J. & Dickens, L., "Integrating Neural Network and Expert Reasoning: an Example," in *Proceedings of AISB, Conference on Developments of Biological Standardization*, 1991.

[HENK94] Henke, Andrea, “Scheduling Space Shuttle Missions: Using Object-Oriented Techniques,” *AI Expert* - Vol. 9, No 3, pp. 16-24 - 1994.

[HILL88] Hill, N. C. and Sartoris, W. L., *Short-term Financial Management*, New York: Macmillan Pub. Co., 1988.

[HIRO93] Hirota, K. (editor), *Industrial Applications of Fuzzy Technology*, Tokyo: Springer-Verlag, 1993.

[HOFF93] Hoffman, Forrest & Tripathi, Vijai S., “A Geochemical Expert System Prototype Using Object-Oriented Knowledge Representation and a Production Rule System,” *Computers & Geosciences*, Vol. 19, No. 1, pp. 53-60 - 1993.

[HONA94] Honavar, V. and Uhr, L. (editors), *Artificial Intelligence and Neural Networks: Steps Toward Principed Integration*, San Diego: Academic Press, 1994.

[HOPF82] Hopfield, J. J., “Neural Networks and Physical Systems with Emergent Collective Computational Abilities,” *Proceedings of the National Academy of Sciences* 79, pp.2554-2558,1982. (Also in [ANDE89] pp. 460-464).

[HORG94] Horgan, J., “Neural Eavesdropping,” *Scientific American*, Vol. 270, No. 5, May, 1994.

[HORN85] Horne, J. C. V.; Dipchand, C. R.; and Hanrahan, J. R., *Fundamentals of Financial Management*, 5th ed., Ontario: Prentice-Hall, 1985.

[HSIE] Hsieh, Chang-tseh, “Some Potential Application of Artificial Neural Systems in Financial Management,” *Journal of Systems Management*, Vol. 44, No. 4, pp. 12-15, April - 1993.

[HUAN95] Huang, S. H. and Zhang, H-C, "Neural-expert Hybrid Approach for Intelligent Manufacturing: a Survey," *Computers in Industry*, Vol. 26, pp. 107-126, 1995.

[HUSH93] Hush, D. R. and Horne, B. G., “Progress in Supervised Neural Networks - What’s New Since Lippman?”, *IEEE Signal Processing Magazine*, Vol. 10, pp. 8-39, 1993.

[JACO86] Jacob, R. J., and Froscher, J. N., “Software Engineering for Rule-Based Systems.” *Proceedings of the Fall Joint Computer Conference* 1st, Dallas, Baltimore: IEEE Computer Society Press, MD, pp. 185-189, 1986.

[JACO88] Jacobs, R. A., “Increased Rates of Convergence Through Learning Rate Adaptation,” *Neural Networks*, Vol. 1, pp. 295-307, 1988.

[JAMS93] Jamshidi, M.; Vadiee, N.; Ross, T. (editors), *Fuzzy Logic and Control: Software and Hardware Applications*, Englewwod,NY: Prentice-Hall, 1993.

[JANG95] Jang, J. R. and Sun, T., “Neuro-Fuzzy Modeling and Control,” *Proceedings of IEEE*, Vol. 83, No. 3, pp. 378-406, 1995.

[KAIN94] Kaindl, H., “Object-Oriented Approaches in Software Engineering and Artificial Intelligence,” *Journal of Object-Oriented Programming*, Vol. 6, No. 8, pp. 38-45, January, 1994.

[KANA86] Kanal, L. N. and Lemmer, J. F. (editors), *Uncertainty in Artificial Intelligence*, Amsterdam: North-Holland, 1986.

[KAND92] Kandel, A., *Fuzzy Expert Systems*, Boca Raton: CRC Press, 1992.

[KAND92b] Kandel, A. and Langholz, G. (editors), *Hybrid Architectures for Intelligent Systems*, Boca Raton: CRC Press, 1992.

[KAND94] Kandel, A. and Langholz, G. (editors), *Fuzzy Control Systems*, Boca Raton: CRC Press, 1994.

[KAND96] Kandel, A.; Pacheco, R.; Martins, A. and Khator, S., “Foundations of Rule-Based Computations in Fuzzy Models,” in *Fuzzy Modelling: Paradigms and Practice*, ed. by W. Pedrycz, Netherlands: Kluwer Academic Pub., 1996.

[KHEB95] Khebbal, S. and Shamhong, D., “Tools and Environments for Hybrid Systems,” in [GOON95].

[KLIR95] Klir, G. and Yuan, B., *Fuzzy Sets and Fuzzy Logic: theory and applications*, Upper Saddle River, NJ: Prentice Hall, 1995.

[KOHO72] Kohonen, T., “Correlation Matrix Memories,” *IEEE Transactions on Computers*, C-21, pp. 353-359, 1972. (Also in [ANDE89] pp. 174-180).

[KOSK92] Kosko, B., “Fuzzy Systems as Universal Approximators,” *IEEE International Conference on Fuzzy Systems*, San Diego, CL, pp. 1153-1162, March 8-12, 1992.

[LACH91] Lacher, R. C.; Coats, P. K.; Sharma, S. C. and Fant, L. F., “A Neural Network for Classifying the Financial Health of a Firm,” *Technical Report* of The Florida State University, Dept. of Computer Science (obtained by personal contact with the author), 1991.

[LAUG89] Laughlin, S., “The Reliability of Single Neurons and Circuit Design,” in *The Computing Neuron*, Ed. by R. Durbin, C. Miall and G. Mitchison, Wokingham: Addison-Wesley, 1989.

[LAWR92] Lawrence, J., *Introduction to Neural Networks and Expert Systems,* Nevada: California Scientific Software, 4a. ed., 1992.

[LEE90] Lee, C. C., “Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Parts I and II,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 2, pp. 404-435,1990.

[LEE91] Lee, S. and Kil, R. M., “A Gaussian Potential Function Network with Hierarchically Self-Organizing Learning,” *Neural Networks*, Vol. 4, pp. 207-224, 1991.

[LEFE92] Lefebvre, Wesley Curtis, *An Object-Oriented Methodology for the Analysis of to Artificial Neural Networks,* Master’s Thesis, University of Florida - 1992.

[LEUN90] Leung, K. S. & Wong, M. H., “An Expert-system Shell Using Structured Knowledge: An Object-Oriented Approach,” *Computer,* Vol. 23, pp.38-47, March - 1990.

[LEV74] Lev, B., *Financial Statement Analysis: A New Approach*, Englewood Cliffs NJ: Prentice-Hall, 1974.

[LEVE44] Levenberg, K., “A Method for the Solution of Certain Non-linear Problems in Least Squares,” *Quarterly of Applied Mathematics*, Vol. 2, pp. 164-168, 1944.

[LEVE44] Levenberg, K., “A Method for the Solution of Certain Non-linear Problems in Least Squares,” *Quarterly of Applied Mathematics*, Vol. 2, pp. 164-168, 1944.

[LIEB90] Liebowitz, J. (editor), *Expert Systems for Business and Management*, Englewood Cliffs, NJ: Prentice -Hall, 1990.

[LIEB93] Liebowitz, Jay, "Roll Your Own Hybrids," *Byte,* Vol. 18, pp. 113-115, July, 1993.

[LIN93] Lin, Mei & Lin, Frank C., “Analysis of Financial Data Using Neural Nets,” *Neural Network Special Report*, pub. by AI Expert, pp.33-37 - 1993.

[LOWE94] Lowe, D., “Radial Basis Function Networks: a Brief Review of the Theory of Radial Basis Function Networks,” Defense Research Agency, Malvern, UK, February 16,1994.

[LOWE95] Lowe, D. G., “Similarity Metric Learning for a Variable-Kernel Classifier,” *Neural Computation*, Vol. 7, No. 1, pp. 72-85, 1995.

[LOWE95b] Lowe, D., "Radial Basis Function Networks," *in Handbook of Brain Theory*, ed. by M. Arbib, Cambridge: MIT Press, 1995.

[MAMD75] Mamdani, E. H. and Assilian, S., “An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller,” *International Journal of Man-Machine Studies*, Vol. 7, No. 1, pp. 1-13, 1975.

[MARK93] Markos, S.; Monostori, L.; Nacsa, J.; and Szöllõsi, G., "Monitoring of the Milling Process Through Neural Network and Fuzzy Techniques,"*in [MEZG93]*, pp.239-250, 1993.

[MARQ63] Marquardt, D. W., “An Algorithm for Least-Squares Estimation of Nonlinear Parameters,” *Journal of Society for Industrial and Applied Mathematics*, Vol. 11, No. 2, pp. 431-441, 1963.

[MART96] Martins, A. R., *Aplicação e Validação de um Sistema Híbrido na Análise de Demonstrativos Financeiros*. PhD Dissertation, Federal Univ. of Santa Catarina, Production Engineering Department, Brazil, 1996.

[MARTK94] Martikainen, T.; Puhalainen, K., and Yli-Olli, P., “On the Industry Effects on the Classification Patterns of Financial Ratios,” *Scandinavian Journal of Management*, Vol. 10, No. 1, pp. 59-68, 1994.

[MAST93] Masters, T., *Practical Neural Network Recipes in C++,* San Diego, CA: Academic Press Inc., 1993.

[MCCL81] McClelland, J. L. and Rumelhart, D., “An Interactive Activation Model of Context Effects in Letter Perception: Part 1. An Account of Basics Findings,” *Psychological Review* 88, pp. 375-407, 1981. (Also in [ANDE89] pp. 404-436).

[MCCU43] McCulloch, W. S. and Pitts, W., “A Logical Calculus of the Ideas Immanent in Nervous Activity,” *Bulletin of Mathematical Biophysics* 5, pp. 115-133, 1943. (Also in [ANDE89] pp. 18-27).

[MCMU79] McMullen, S. Y., *Financial Statements: Form, Analysis, and Interpretation*, 7th Ed., Homewood, IL: Richard D. Irwin Inc., 1979.

[MEDS92] Medsker, Larry & Bailey, David, "Models and Guidelines for Integrating Expert Systems and Neural Networks," *In in [KAND92]*, pp. 154-171, 1992.

[MEDS94] Medsker, L. R., *Hybrid Neural Network and Expert Systems,* Norwell, MA, USA: Kluwer Academic Publishers, 1994.

[MEDS94b] Medsker, L. R. and Turban, E.- "Integrating Expert Systems and Neural Computing for Decision Support," *Expert Systems with Applications,* Vol. 7, No. 4, Oct-Dez, 1994.

[MEDS95] Medsker, L., *Hybrid Intelligent Systems,* Boston: Kluwer Academic Publishers, 1995.

[MEND93] Mendelsohn, Lou, "Neural Net Applications for Market Analysis," *AI in Finance Special Report*, pub. by AI Expert, pp. 60-65, October - 1993.

[MEND95] Mendel, J. M., "Fuzzy Logic Systems for Engineering: A Tutorial," *Proceedings of IEEE*, Vol. 83, No. 3, pp. 345-377, March, 1995.

[MENG42] Menger, K., "Statistical Metrics," *Proceedings of Nat. Acad. Sci.*, USA, Vol. 28, pp. 535-537, 1942.

[MEZG93] Mezgár, L and Bertók, P., *Knowledge Based Hybrid Systems*, Amsterdam: North Holland-Elsevier Science Pub., 1993.

[MICC86] Micchelli, C. A., "Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions," *Constructive Approximation*, Vol. 2, pp. 11-22, 1986.

[MINS69] Minsky, M. and Papert, S., *Perceptrons,* Cambridge: MIT Press, 1969

[MINS75] Minsky, M., "A Framework for Representing Knowledge," In *The Psychology of Computer Vision*," ed. by P. Winston, pp. 211-277, New York: McGraw-Hill, 1975.

[MINS88] Minsky, M.; "Connectionist Models and Their Prospects," in *Connectionist Models and Their Implications: Readings From Cognitive Science*, Ed. by David Waltz and Jerome A. Feldman, Norwood, NJ: Ablex Pub. Co., pp. vii-xvi, 1988.

[MONT94] Montague, P. R.; Gancayco, C. D.; Winn, M. J.; Marchase, R. B.; and Friedlander, M. J., "Role of NO Production in NMDA Receptor-Mediated Neurotransmitter Release In Cerebral Cortex," *Science*, Vol. 263, No. 5149, pp. 973-979, February, 1994.

[MONTE94] Montenegro, Fernando & Pacheco, Roberto C. S., *Orientação a Objetos em C++*, Rio de Janeiro: Ed. Ciência Moderna, 1994.

[MOOD89] Moody, J. and Darken, C., "Fast Learning in Networks of Locally-Tuned Processing Units," *Neural Computation* 1, pp. 281-294, 1989.

[MUI90] Mui, C. K., Haffel, C. and Courtis, L. "The Financial Statement Analyzer", in [LIEB90] pp. 127-142, 1990.

[MÜLL90] Müller, B. and Reinhardt, J., *Neural Networks: An Introduction* - Physics of Neural Networks Series, Berlin: Springer-Verlag, 1990.

[MUNA94] Munakata, T. and Jani, Y., "Fuzzy Systems: An Overview," *Communications of ACM*, Vol. 37, No. 3, March, 1994.

[NEAP90] Neapolitan, R. E., *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*, New York: John Wiley & Sons, 1990.

[NEWE57] Newell, A. and Shaw, J. C., "Programming the Logic Theory Machine," *Proceedings of the Western Joint Computer Conference*, Vol. 15, pp. 230-240, 1957.

[NEWE63] Newell, A., and Simon, H. A., "GPS, a Program that Simulates Human Thought," in *Computers and Thought*, ed. by E. A. Feigenbaum and J. Feldman, New York: McGraw-Hill, 1963.

[NEWE72] Newell, A. and Simon, H. A., *Human Problem Solving*, Elgelwood Cliffs, NJ: Prentice-Hall, 1972.

[NIYO94] Niyogi, P. and Girosi, F., “On the Relationship Between Generalization Error, Hypothesis Complexity, and Sample Complexity for Radial Basis Functions,” *A.I. Memo No. 1467*, M.I.T., Artificial Intelligence Lab., (ftp:publications.ai.mit.edu/ai-publications/1994/AIM-1467.ps.Z), 1994.

[NOTT91] Nottola, C.; Condamin, L.; and Naïm, P., "On the use of Hard Neural Networks for Symbolic Learning Application to Company Evaluation," in *Proceedings of the International Joint Conference on Neural Networks IJCNN-91*, Seattle, pp. 509-522, 1991.

[ODOM93] Odom, M. and Sharda, R.; “A Neural Network Model for Bankruptcy Prediction,” in [TRIP93], pp. 177-185, 1993.

[ORFA90] Orfanidis, S. J., “Fast Neural Nets with Gram-Schmidt Orthogonalization,” in *Proceedings of the International Joint on Artificial Neural Networks 1990 (IJCNN-90)*, Washington, N.J.: Lawrence Associates, 1990.

[OSTE92] Osteryoung, J.; Constand, R. and Nast, D., Financial Ratios in Large Public and Small Private Firms, *Journal of Small Business Management,* Vol. 30, No. 3, July 1992.

[PACH91] Pacheco, Roberto C. S., *Tratamento e Imprecisão em Sistemas Especialistas,* Mater’s Thesis, Universidade Federal de Santa Catarina UFSC, Programa de Pós-graduação em Engenharia de Produção, Florianópolis, 1991.

[PACH95] Pacheco, R.; Martins, A.; Weber, R.; Barcia, R. M.; and Khator, S., "A Hybrid Intelligent System to Diagnose and Indicate Solutions to Financial Problems," in *Proceedings of the First International Congress of Industrial Engineering and XV National Congress of Production Engineering (ENEGEP)* Vol. III, pp. 1858-1862, São Carlos SP, Brazil, September, 1995.

[PACH96] Pacheco, R.; Martins, A.; Barcia, R.; Khator, S.; “A Hybrid Intelligent System Applied to Financial Statement Analysis,” *submitted to Fifht IEEE International Conference on Fuzzy Systems (FUZZ-96-IEEE)*, to be held in New Orleans, USA, Sept. 8-11, 1996.

[PACH96b] Pacheco, R.; Martins, A.; Barcia, R.; Khator, S.; “Financial Diagnosis Through a Hybrid Intelligent System,” *submitted to EUFIT’96 - The Fourth Eurpean Congress on Intelligent Techniques and Soft Computing*, to be hel in Aachen, Germany, Sept. 2-5, 1996.

[PEAR86] Pearl, J. “One the Evidential Reasoning in a Hierarchy of Hypotheses,” *Artificial Intelligence*, Vol. 28, pp. 9-15, 1986.

[PEAR88] Pearl, J., *Probabilistic Reasoning In Intelligent Systems: Networks Of Plausible Inference*, San Mateo, CA.: Morgan Kaufmann Publishers, 1988.

[PENF75] Penfield, W.; *The Mystery of the Mind: A Critical Study of Consciouness and the Human Brain*, Princeton, NJ: Princeton University Press, 1975.

[PLAC56] Place, U. T.; “Is Consciousness a Brain Process?”, *British Journal of Psychology* 47, pp. 44-50, 1956.

[POGG89] Poggio, T. and Girosi, F., “A Theory of Networks for Approximation and Learning,” *A.I. Memo No.* 1140, CBIP Paper No. 31, MIT, AI Lab., (ftp://128.52.38.22l-publications\1000-1499\aim-1140.ps.z), 1989.

[POGG90] Poggio, T. and Girosi, F., “Networks for Approximation and Learning,” *Proceedings of the IEEE*, Vol. 78, No. 9, pp. 1481-1497, Sept., 1990

[POWE87] Powel, M. J. D., “Radial Basis Functions for Multivariable Interpolation: a review,” in *Algorithms for Approximation: Proceedings of the IMA Conference on Algorithms for the Approximation of Functions and Data,* in Shrivenham, July 1985, edited by

J.C. Mason and M.G. Cox., Oxford, New York: Clarendon-Oxford University Press, 1987.

[POWE88] Powell, M. J. D., "Radial Basis Function Approximations to Polynomials," in *Numerical Analysis, 1987 - Proceedings of the 12th Dundee Biennial Conference on Numerical Analysis,* University of Dundee, June 23-26, 1987, ed. by D.F. Griffiths and G.A. Watson, New York : Wiley, 1988.

[QUIL68] Quillian, R., "Semantic Memory," in *Semantic Information Processing*, Ed. by M. Minsky, Cambridge, MA: MIT Press, 1968.

[RABE92] Rabelo, L. C., "A Hybrid Neural and Symbolic Processing Approach to Flexible Manufacturing Systems Scheduling," *in [KAND92]*, pp. 379-405, 1992.

[RAGH93] Raghupathi, W.; Schkade, L. L.; and Raju, B. S.; "A Neural Network Approach to Bankruptcy Prediction," in [TRIP93], pp. 141-158, 1993.

[RAHI93] Rahimian, E.; Seema, S.; Thammachote, T. and Rajiv, V. ; "Bankruptcy Prediction by Neural Network," in [TRIP93], pp. 159-176, 1993.

[RAM90] Ram, Sudha & Ram, Sundaresan, "Screening Financial Innovations: An Expert System Approach," *IEEE Expert*, Vol. 5, No. 9, pp. 20-27, August - 1990.

[RAN91] Ran, Alexander & Darjalainen, Matti, "High Level Object-Oriented Programming Environment for Efficient Neural Computations," *Proceedings of the 1991 International Conference on Artificial Neural Networks* (ICANN-1991), Espoo, Finland, June 24-28 - 1991.

[RICH90] Rich, Elaine, "Expert Systems and Neural Networks Can Work Together, " *IEEE Expert,* Vol. 5, October, 1990.

[RING88] Ringlad, G. A. and Duce, D. A. ed. *Approaches to Knowledge Representation - an Introduction*, London: John Wiley & Sons Inc. - Research Studies Press - 1988.

[RODR96] Rodriguez, Alejandro Martins & Luna, Paulo de Tarso Mendes, *Capital de Giro: Um Enfoque Empresarial,* to be published, 1996.

[ROSE58] Rosenblatt, F., "The Perceptron: A Probalistic Model for Information Storage and Organization in the Brain," *Psychological Review* 65: pp. 386-408 - 1958

[RUME86a] Rumelhart, D. E. & McClelland, J. L. (editors), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* Vols. 1-2, Cambridge: MIT Press, 1986.

[RUME86b] Rumelhart, D. E.; Hinton, G. E. and Williams, R. J., "Learning Internal Representations by Error Propagation", in [RUME86a] pp. 318-362, 1986 (Also in [ANDE89] pp. 673-695).

[RUME86c] Rumelhart, D. E.; Hinton, G. E. and Williams, R. J., "Learning Representations by Back-propagating Errors", *Nature*, Vol. 323, pp. 533-536, 1986 (Also in [ANDE89] pp. 696-699).

[RYCH91] Rychlak, J. F.; *Artificial Intelligence and Human Reason: A Teleological Critique*, New York: Columbia University Press, 1991.

[SALM94] Salmi, T. and Martikainen, T., "A Review of the Theoretical and Empirical Basis of Financial Ratio Analysis," *Finnish Journal of Business Economics*, Vol. 4, 1994. (available at: ftp://garbo.uwasa.fi/pc/research/farewa12.zip).

[SCHO93] Schofeld & Associates, *IRS Corporate Financial Ratios*, Evanston, ILL:Schonfeld & Associates, 7th Edition, 1993.

[SCHU94] Schuman, E. M. and Madison, D.V., "Locally Distributed Synaptic Potentiation in the Hippocampus", *Science*; Vol. 263, No. 5146, pp. 532-537, January, 1994.

[SCHW91] Schwartz, T. J., The Schwartz Associates, Mtn. View, California, 1991.

[SCOT91] Scott, A. C., Clayton, J. E. and Gibson, E. L., *A Practical Guide to Knowledge Acquisition*. Reading, MA: Addison-Wesley, 1991.

[SENJ93] Senjen, R.; Beler, M.; Leckie, C, and Rowles, C., "Hybrid Expert Systems for Monitoring and Fault Diagnosis," in *Proceedings of 9th IEEE Conference on Artificial Intelligence for Applications*, Orlando, FL, March 1-5, pp. 235-241, 1993

[SHAF76] Shafer, Glenn, *A Mathematical Theory of Evidence*, Princeton, NJ: Princeton University Press, 1976.

[SHER92] Sherstinsky, A.; and Picard, R. W., "On Training Gaussian Radial Basis Functions for Image Coding," *MIT Media Laboratory Vision and Modeling Group Technical Report #188*, 1992.

[SHOR76] Shortliffe, E. H., *Computer-Based Medical Consultations: MYCIN*, New York: American Elsevier, 1976.

[SMAR59] Smart, J. J. C.; "Sensations and Brain Processes.", *Philosophical Review* 68, pp. 141-156, 1959.

[SONG94] Song, Q. and Bortolan, G, "Some Properties Of Defuzzification Neural Networks," *Fuzzy Sets and Systems*, Vol. 61, No. 1, pp. 83-89, 1994.

[SOWA91] Sowa, J. F. (editor), *Principle of Semantic Networks: Exploration in the Representation of Knowledge*, San Mateo: Morgan Kaufmann Pub., 1991.

[STEF86] Stefik, M. and Bobrow, G., "Object Oriented Programming: Themes and Variations," *The AI Magazine*, pp. 40-62, Vol. 6, No. 4, Winter 1986.

[SUN95] Sun, R. and Bookman, L. A. (editors), *Computational Architectures Integrating Neural and Symbolic Processes: A Perspective on the State of the Art*, Boston: Kluwer Academic Pub., 1995.

[TAMA78] Tamari, M., *Financial Ratios: Analysis and Prediction*, London: Paul Elek, 1978.

[TELL89] Tello, Ernst, *Object-Oriented Programming for Artificial Intelligence: A Guide to Tools and Systems Design,* Addison-Wesley, USA - 1989.

[TERA94] Terano, T.; Asai, K. and Sugeno, M., *Applied Fuzzy Systems*, Boston: AP Professional (Academic Press, Inc.), 1994.

[THOR93] Thorson, M., Warthman, F., and Holler, M., "A Neural-Network Audio Synthesizer," *Dr. Doobb's Journal*, Vol. 18, Issue 2, #197, February, pp. 50-55, 58-64, 1993.

[TOU74] Tou, J. T. and Gonzalez, R. C., *Pattern Recognition Principles*, Reading, MA: Addison-Wesley, 1974.

[TOWE93] Towell, G. G. and Shavlik, J. W., "Extracting Refined Rules from Knowledge-Based Neural Networks," *Machine Learning*, Vol. 13, No. 1, pp. 71-101, 1993.

[TOWE94] Towell, G. G. and Shavlik, J. W., "Knowledge-Based Artificial Neural Networks," *Artificial Intelligence*, Vol. 70, No. 1-2, pp. 119-165, 1994.

[TRIP93] Trippi, R. R.; and Turban, E. (editors); *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance*, Chicago: Probus Pub. Co., 1994.

[WANG92] Wang, L-X., "Fuzzy systems are universal approximators," *IEEE International Conference on Fuzzy Systems*, San Diego, CL, pp. 1163-1170, March 8-12, 1992.

[WATE78] Waterman, D. A. and Hayes-Roth, F., *Pattern-directed Inference Systems*. New York: Academic Press, 1978.

[WATE86] Waterman, D. A., *A Guide to Expert Systems*, Reading, MA: Addison-Wesley, 1986.

[WEBE93] Weber, Rosina de O., *Sistema Especialista Difuso Para Análise de Crédito.* Master's Thesis, Universidade Federal de Santa Catarina UFSC, Programa de Pós-graduação em Engenharia de Produção, Florianópolis, 1993.

[WELS77] Welsh, G. A. and Anthony, R. N., *Fundamentals of Financial Accounting*, 2nd. Ed. Homewood, IL: Richard D. Irwin Inc., 1977.

[WERB74] Werbos, P. J., “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.” PhD Thesis, Harvard University, Cambridge, MA, 1974.

[WERB94] Werbos, P. J., *The Roots Of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting,*” New York : J. Wiley & Sons, 1994.

[WETT92] Wettschereck, D. and Dietterich, T., “Improving the Performance of Radial Basis Function Networks by Learning Center Locations,” In *Advances In Neural Information Processing Systems 4*, ed. by by J. E. Moody, S. J. Hanson, and R. P. Lippman, San Mateo, CA : Morgan Kaufmann, 1992.

[WIDR60] Widrow, B. and Hoff, M. E., “Adaptative Switching Circuits,” *IRE WESCON Convention Record*, New York: IRE, pp. 94-104, 1960. (Also in [ANDE89] pp. 126-134).

[WILS94] Wilson, R. L. and Sharda, R., “Bankruptcy Prediction Using Neural Networks,” *Decision Support Systems*, Vol. 11, No. 5, pp. 545-557, 1994.

[YAGE83] Yager, R., “On the Implication Operator in Fuzzy Logic,” *Information Sciences*, Vol. 31, pp. 141-164, 1983.

[YAGE85] Yager, R., “Fuzzy Set Aggregation Connectives,” *Information Sciences*, Vol. 36, pp. 85-121, 1985.

[YAGE93] Yager, R. and Filev, D. P., “On the Issue of Defuzzification and Selection Based on a Fuzzy Set.” *Fuzzy Sets and Systems*, Vol. 55, No. 3, pp. 255-272, 1993.

[YAGE94] Yager, R. and Filev, D. P., *Essentials of Fuzzy Modeling and Control*, New York: John Wiley & Sons, Inc., 1994.

[YAGE94b] Yager, R. R., Kacprzyk, J. and Fedrizzi, M. (editors), *Advances In The Dempster-Shafer Theory Of Evidence*, New York : J. Wiley, 1994.

[YAGE95] Yager, R. R., *Theoretical Aspects of Fuzzy Control*, New York : J. Wiley, 1995.

[YELL93] Yelland, P. M., “Object Oriented Knowledge Representation, ” *BT Technology J.*, Vol. 11, No 3, pp. 41-51, July - 1993.

[YIU92] Yiu, Kenneth L. K. & Kong, Andy W. K., “Choosing the Correct Expert System Development Method for Financial Decision-Making,” *Journal of Systems Management*, Vol. 43, No. 11, pp. 16-19, 30-31, 37-38, 40-43, November - 1993.

[YOON94] Yoon, Y.; Guimaraes, T.; and Swales, G.; "Integrating Artificial Neural Networks with Rule-based Expert Systems," *Decision Support Systems* Vol. 11, pp. 497-507, 1994.

[ZADE65] Zadeh, L. A., “Fuzzy Sets,” *Information and Control*, 8, pp. 338-353, 1965.

[ZADE73] Zadeh, L. A., “Outline of a New Approach to the Analysis of Complex Systems and Decision Process,” *IEEE Transactions on Systems, Man and Cybernetics*, 3, 28-44, 1973.

[ZADE75] Zadeh, L. A., “Fuzzy Logic and Approximate Reasoning (In Memory of Grigore Moisil),” *Synthese*, 30, pp. 407-428, 1975.

[ZADE78] Zadeh, L. A., “Fuzzy Sets as a Basis for a Theory of Possibility,” *Fuzzy Sets and Systems*, Vol. 1, pp. 3-28, 1978.

[ZADE79] Zadeh, L. A., “A Theory of Approximate Reasoning,” in *Machine Intelligence 9*, ed. by J. E. Hayes, D. Michie and L. I. Mikulich, New York: John Wiley & Sons, 1979.

[ZADE83] Zadeh, L. A., “The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems,” *Fuzzy Sets and Systems*, Vol. 11, pp. 199-227, 1983. (Also in [GUPT85], pp. 3-31).

[ZADE92] Zadeh, L. A. and Kacprzyk, J. (editors), *Fuzzy Logic for the Management of Uncertainty*, New York: John Wiley & Sons, 1992.

[ZADE94] Zadeh, L. A., “Fuzzy Logic, Neural Networks, and Soft Computing,” *Communications of the ACM*, Vol. 37, No. 3, pp. 77-84, March, 1994.

[ZAHE93] Zahedi, Fatemah, *Intelligent Systems for Business: an Expert Systems with Neural Networks*, Belmont, CA: Wadsworth pub. co., 1993.

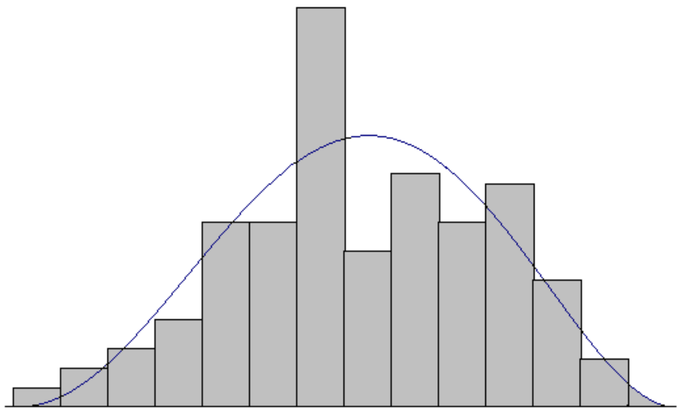
7

Appendices

7.1. Appendix A:

Financial Ratio Distributions

Financial Ratio 2:
Net Margin



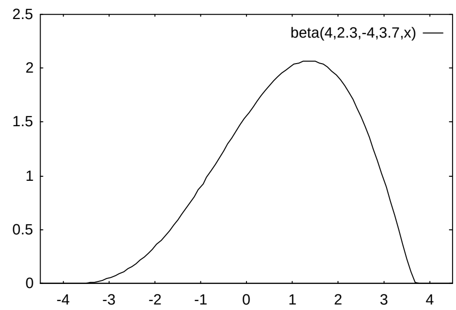
BEST FIT SUMMARY	
Function	Sq Error
Beta	0.0110
Weibull	0.0115
Normal	0.0119
Triangular	0.0120
Erlang	0.0139
Gamma	0.0143
Lognormal	0.0197
Uniform	0.0400
Exponential	0.0702

Final Distribution:

mean = 0.5936; StdDev = 1.31; min = -2.81; max = 3.21; histogram range: -3 to 3.82
Best Fitting: Beta Distribution: (3.31, 2.99);
Sq. Error: 0.01099

Original Distribution:

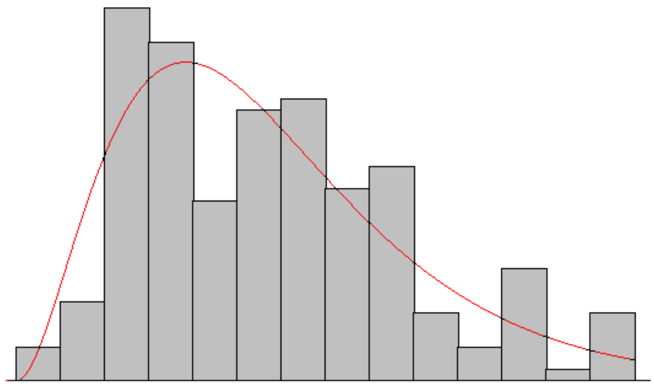
Bucksmaster's U shape
Beta Distribution: beta(4,2.3);
Chosen interval [-4,3.7]



Ratio Limits (Profit and Loss) According to the firm size [SCHO93]:

All		1M to 999.99M		1MM to 24.99MM		25MM to 999.99MM		100MM and over	
1.9	-1.7	2.2	-2.6	1.5	-0.3	2.6	0.6	2	-1.3

Financial Ratio 3:
Cash Conversion Cycle



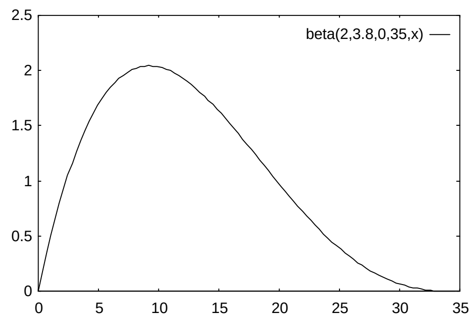
BEST FIT SUMMARY	
Function	**Sq Error**
Erlang	0.0104
Triangular	0.0104
Gamma	0.0108
Lognormal	0.0109
Weibull	0.0116
Beta (1.73,2.48)	0.0136
Normal	0.0168
Uniform	0.0370
Exponential	0.0499

Final Distribution:

mean = 14.1; StdDev = 6.9; min =1.91; max = 32.54; histogram range: 1 to 33
Best Fitting: Erlang; Sq. Error: 0.01035

Original Distribution:

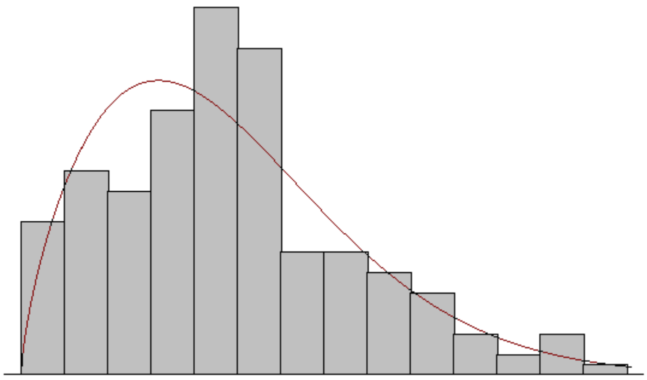
Bucksmaster's Reverse J
Beta Distribution: beta(2, 3.8);
Chosen interval: [0,35]



Ratio Limits (Profit and Loss) According to the firm size [SCHO93]:

All		1M to 999.99M		1MM to 24.99MM		25MM to 999.99MM		100MM and over	
15.8	2.8	14.8	4.7	6.0	4.2	3.5	3.6	20.9	

Financial Ratio 4:
Sales to Working Capital



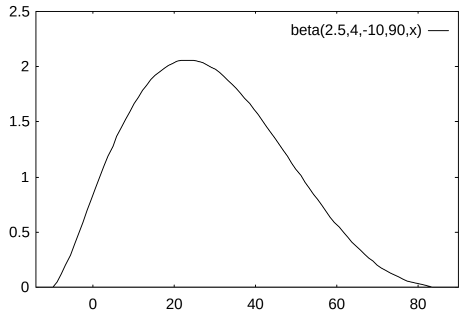
BEST FIT SUMMARY	
Function	Sq Error
Weibull	0.0089
Normal	0.0089
Beta (1.51,2.86)	0.0096
Gamma	0.0126
Erlang	0.0133
Triangular	0.0175
Lognormal	0.0242
Exponential	0.0364
Uniform	0.0394

Final Distribution:

mean = 28.6; StdDev = 15.8; min = 2.26; max = 78.12; range: 2 to 79
Best Fitting: Weibull; Sq. Error: 0.008888

Original Distribution:

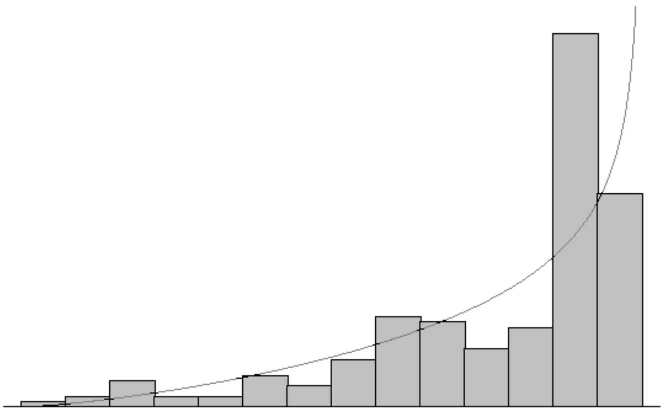
Bucksmaster's J shape
Beta Distribution: beta(2.5, 4);
Chosen interval: [-10,90]



Ratio Limits (Profit and Loss) According to the firm size [SCHO93]:

All		1M to 999.99M		1MM to 24.99MM		25MM to 999.99MM		100MM and over	
24.6	-122	15.1	54.5	31.1	71.7	41.6	62.1	23.1	-54.6

Financial Ratio 5:
T to Working Investment (TWI)



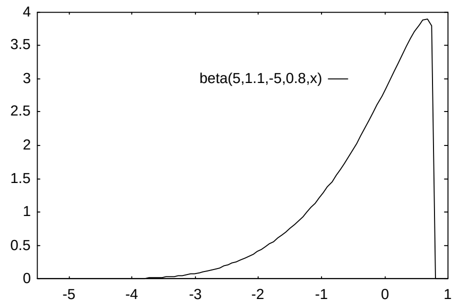
BEST FIT SUMMARY	
Function	**Sq Error**
Beta	0.0514
Weibull	0.0754
Normal	0.0817
Triangular	0.0886
Erlang	0.1003
Gamma	0.1005
Lognormal	0.1126
Uniform	0.1176
Exponential	0.1607

Final Distribution:

mean = -0.6063; StdDev = 1.47; min = -5.6; max = 0.80; histogram range: -6 to 1
Best Fitting: Beta (2.33, 0.6927); Sq. Error: 0.05137

Original Distribution:

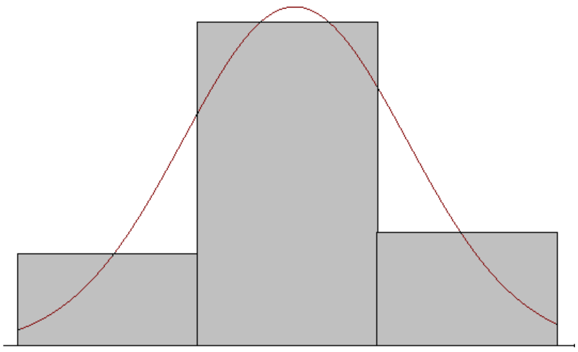
Bucksmaster's J shape
Beta Distribution: beta(5,1.1);
Chosen interval: [-5,1]



Ratio Limits (Profit and Loss) According to the firm size [SCHO93]:

All		1M to 999.99M		1MM to 24.99MM		25MM to 999.99MM		100MM and over	
-0.15	-3.13	0.35	0.16	0.39	-0.18	0.25	1) -0.24	-0.26	-2.19

Financial Ratio 6:
T/|WI| trend



BEST FIT SUMMARY	
Function	**Sq Error**
Normal	0.0013
Weibull	0.0027
Triangular	0.0049
Beta	0.0063
Erlang	0.0073
Gamma	0.0093
Lognormal	0.0214
Uniform	0.1156
Exponential	0.2287

Final Distribution:

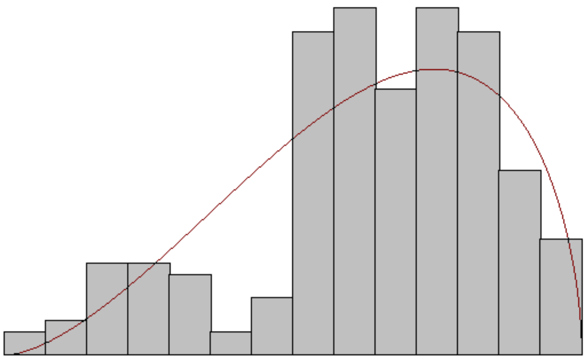
mean = 0.04; StdDev = 0.6248; min = -1; max = 1; histogram range: -1 to 1
Best Fitting: norm(0.04, 0.6232); Sq. Error: 0.001273

Original Distribution:

normal Distribution: normal(0,1); Chosen interval: [-1,1]

Obs: this ratio does not have sample data in [SCHO93]. The decision was taking the trend as a normal distribution with average 0, that is, in the average, the firms have a stable trend regarding the ratio T/|WI|.

Financial Ratio 7: EBIT to Interest Charges



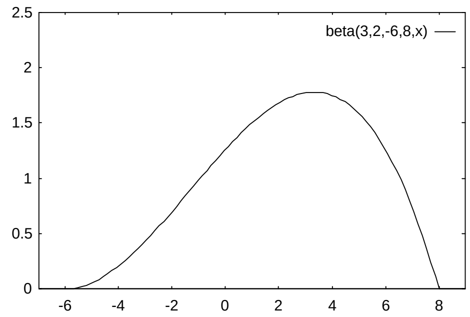
BEST FIT SUMMARY	
Function	**Sq Error**
Beta	0.0127
Normal	0.0147
Weibull	0.0153
Triangular	0.0285
Erlang	0.0298
Gamma	0.0306
Uniform	0.0404
Lognormal	0.0416
Exponential	0.0777

Final Distribution:

mean = 2.9; StdDev = 2.99; min = -5.34; max = 7.82; histogram range: -6 to 8
Best Fitting: Beta: (2.67, 1.57); Sq. Error: 0.01274

Original Distribution:

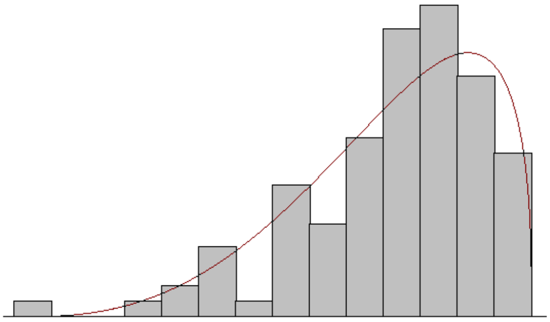
Bucksmaster's J shape
Beta Distribution: beta(3,2);
Chosen interval: [-6,8]



Ratio Limits (Profit and Loss) According to the firm size [SCHO93]:

All		1M to 999.99M		1MM to 24.99MM		25MM to 999.99MM		100MM and over	
3.1	0.4	6.5	-3.1	5.2	0.7	9.0	2.0	2.6	0.7

Financial Ratio 8:
Debt to Equity



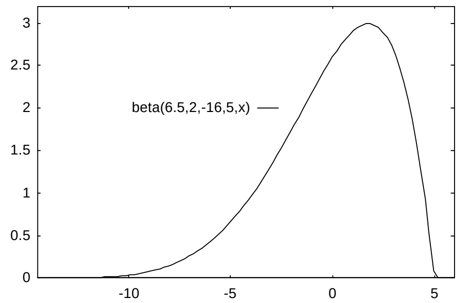
BEST FIT SUMMARY	
Function	**Sq Error**
Beta	0.0071
Weibull	0.0010
Normal	0.0149
Erlang	0.0286
Gamma	0.0288
Triangular	0.0294
Lognormal	0.0394
Uniform	0.0646
Exponential	0.1073

Final Distribution:

mean = 0.7782; StdDev = 2.86;
min = -10.11; max = 4.98; histogram range: -11 to 5
Best Fitting: Beta(3.77, 1.39); Sq. Error: 0.007098

Original Distribution:

Bucksmaster's J shape
Beta Distribution: beta(6.5,2);
Chosen interval: [-16,5]



Ratio Limits (Profit and Loss) According to the firm size [SCHO93]:

All		1M to 999.99M		1MM to 24.99MM		25MM to 999.99MM		100MM and over	
1.8	-6.2	1.2	-11.5	1.5	3.9	1.0	2.4	1.9	-5.2

7.2. Appendix B:

Samples Used to Train the Neural Networks

No.	R1	R2	R3	R4	R5	R6	R7	R8	Prof	STL	Debt	FA
1	-0.22	-0.68	8.22	47.70	-2.05	0	-5.00	4.01	1	1	0	0
2	2.41	1.07	11.10	56.21	-0.21	-1	2.80	0.40	0	1	0	0
3	2.37	1.11	10.05	40.83	-0.78	-1	5.70	-2.82	0	1	1	0
4	4.13	2.19	12.86	30.79	-3.20	-1	5.80	0.54	0	1	0	0
6	2.07	0.71	12.22	13.47	0.05	-1	2.33	1.33	0	1	0	0
7	3.50	0.80	13.05	21.69	-2.70	0	2.10	-1.47	0	1	1	0
10	2.42	-1.26	8.58	36.90	-4.21	-1	-2.12	-0.32	0	1	1	0
11	-0.20	-0.70	14.54	42.26	-2.22	0	-4.53	3.19	1	1	0	0
12	1.34	-1.68	1.91	20.47	0.23	0	-3.50	-1.15	0	0	1	0
13	2.36	0.70	17.66	53.54	0.62	0	1.75	3.94	0	1	1	0
14	6.37	2.80	5.70	34.14	-1.45	0	7.20	0.88	0	1	0	0
15	2.31	-1.74	9.27	24.79	0.69	0	-2.21	-4.55	0	0	1	0
16	4.20	2.33	7.07	40.32	0.74	0	5.90	0.10	0	0	0	1
19	2.30	0.80	6.48	17.96	-1.73	0	5.49	-1.99	0	1	1	0
21	3.90	1.90	6.54	15.56	-1.22	0	4.8	1.25	0	1	0	0
22	2.04	-1.37	27.73	22.15	-0.45	-1	-3.97	2.49	0	1	1	0
23	3.26	1.40	7.42	37.62	0.79	-1	4.23	2.67	0	0	0	1
24	1.28	0.60	9.75	28.02	0.60	0	2.07	0.04	0	0	1	0
25	2.30	0.43	13.91	16.81	-2.47	0	2.07	4.37	0	1	0	0
26	-0.79	-1.82	18.23	16.53	-5.03	0	-5.34	3.18	1	1	0	0
27	5.35	2.05	17.79	28.25	0.43	0	5.26	-2.05	0	0	1	0
29	5.2	1.70	10.73	33.97	-1.03	-1	4.24	0.47	0	1	0	0
30	1.85	0.87	19.54	10.63	0.30	-1	2.09	-0.07	0	0	1	0
31	2.01	-1.12	7.60	34.62	0.40	0	-2.39	-1.45	0	0	1	0
34	4.35	1.98	8.33	27.78	-0.33	0	6.97	0.10	0	1	0	0
36	4.6	2.12	14.98	40.82	0.65	0	7.56	-2.41	0	0	1	0
37	5.17	2.50	17.61	61.68	0.23	0	6.3	-0.89	0	0	1	0
38	2.28	0.80	13.55	24.44	0.33	0	4.89	-0.53	0	0	1	0
39	-0.11	-0.54	20.96	32.33	-1.19	-1	-1.49	-3.71	1	0	1	0
42	2.97	0.10	9.57	27.92	0.24	0	3.74	-1.36	0	0	1	0
43	4.23	2.12	13.04	29.32	0.60	0	5.23	2.98	0	0	0	1
44	1.68	-0.86	12.13	16.88	-0.56	0	-4.54	0.05	0	1	1	0
45	5.26	2.32	9.66	24.52	0.41	1	6.00	1.53	0	0	0	1
46	3.61	1.93	23.35	43.91	-0.75	-1	5.35	2.70	0	1	0	0
47	4.23	2.52	21.26	19.98	-1.76	0	7.03	2.55	0	1	0	0
48	2.19	0.35	7.03	27.87	-2.16	1	4.04	-6.84	0	1	1	0
50	3.82	-0.99	12.95	5.03	-1.73	0	2.23	1.17	0	1	0	0
54	6.01	2.80	17.02	23.95	0.38	1	7.20	-0.07	0	0	1	0
55	3.60	-0.60	7.95	32.01	0.78	0	1.84	-9.94	0	0	1	0
56	3.34	-0.91	21.67	34.18	0.59	0	-3.18	-5.60	0	0	1	0
58	5.02	2.23	7.56	31.15	-0.10	0	6.10	-2.58	0	1	1	0
59	3.20	-2.65	6.36	8.91	-1.76	0	-3.20	3.72	0	1	1	0
61	1.43	-0.39	20.64	17.66	-0.13	1	1.52	1.48	0	0	1	0
63	4.21	2.31	8.70	22.18	0.56	0	6.36	-4.44	0	0	1	0
64	0.35	-0.22	11.70	32.12	-4.80	1	-3.53	4.76	0	1	1	0
65	2.99	1.75	15.48	4.01	-1.53	0	4.23	2.27	0	1	0	0
66	3.24	1.23	5.71	31.96	-1.52	-1	3.80	-3.18	0	1	1	0
67	3.74	2.09	14.90	60.26	-1.98	0	5.32	-6.34	0	1	1	0
69	5.20	2.70	12.78	78.12	0.05	0	6.90	-4.79	0	0	1	0
70	2.06	0.58	15.27	40.83	-1.93	0	3.20	2.91	0	1	0	0
71	5.56	1.10	16.27	9.89	-2.03	-1	6.72	-7.11	0	1	1	0
72	2.70	1.55	8.38	16.63	-1.28	-1	4.26	4.09	0	1	0	0
73	3.45	1.69	14.93	8.15	0.31	0	4.2	0.7	0	0	0	1
74	2.79	1.55	6.51	67.52	-0.75	-1	3.76	-1.4	0	1	1	0
75	3.01	0.97	19.92	2.37	-1.35	0	3.25	2.62	0	1	0	0
76	0.52	-0.06	13.59	9.66	-3.62	-1	2.26	1.87	1	1	0	0
77	1.20	-2.81	17.01	24.7	-3.39	-1	-4.04	3.24	1	1	0	0
78	5.48	2.90	8.36	27.05	0.20	0	7.20	0.36	0	0	0	1
79	3.03	1.23	7.74	13.38	-0.94	-1	5.29	2.14	0	1	0	0
80	3.57	1.84	20.23	40.13	0.59	0	5.00	4.80	0	0	1	0

No.	R1	R2	R3	R4	R5	R6	R7	R8	Prof	STL	Debt	FA
81	6.30	3.21	9.92	19.29	0.51	0	7.82	3.21	0	0	0	1
82	4.87	2.25	14.35	31.81	0.03	0	5.60	1.61	0	0	0	1
85	3.66	1.85	14.75	61.40	-1.23	0	6.26	1.26	0	1	0	0
88	3.31	1.50	20.12	31.61	0.07	0	4.24	2.90	0	0	0	1
89	3.13	1.78	5.79	34.64	-1.33	0	4.93	0.89	0	1	0	0
90	2.73	-1.54	16.86	24.71	-4.12	1	-2.60	2.07	0	1	0	0
91	2.54	0.92	7.93	8.88	0.30	1	5.30	2.19	0	1	0	0
93	5.47	2.78	20.01	72.34	0.65	0	7.20	0.28	0	0	0	1
94	2.34	1.24	7.08	28.32	-3.16	-1	5.63	4.81	0	1	1	0
97	1.96	1.02	28.05	6.15	-1.94	0	3.29	4.50	0	1	1	0
98	5.20	3.04	29.63	16.23	0.60	0	7.47	-4.62	0	0	1	0
99	4.70	2.92	18.14	33.94	0.33	0	7.45	-10.11	0	0	1	0
100	2.81	1.05	7.68	10.54	-1.72	0	3.04	1.67	0	1	0	0
101	4.65	0.93	8.43	10.88	-0.18	-1	6.00	1.21	0	1	0	0
103	2.00	0.21	6.48	7.38	-0.55	-1	5.49	-0.87	0	1	1	0
106	1.06	-0.04	13.09	2.26	-1.11	0	2.11	-5.70	0	1	1	0
107	4.81	2.80	5.33	9.86	0.38	0	6.78	-2.69	0	0	1	0
109	4.93	1.34	9.35	50.06	0.40	0	5.83	1.15	0	0	0	1
110	4.35	1.50	17.23	23.52	-2.42	1	4.52	-4.47	0	1	1	0
111	1.25	-1.50	3.42	25.56	0.35	0	2.20	1.43	1	0	0	0
113	3.69	-0.50	12.47	25.61	0.14	0	1.64	4.14	0	0	1	0
116	4.67	2.64	4.36	26.61	0.27	1	5.65	-2.41	0	0	1	0
117	2.20	1.07	20.31	72.92	0.79	1	5.09	0.02	0	0	0	1
118	2.55	-0.49	22.7	43.62	-3.83	0	2.14	2.08	0	1	0	0
119	3.03	0.12	8.15	14.5	0.63	0	4.65	2.95	1	0	0	0
120	3.06	-0.32	11.65	43.49	-1.98	1	4.09	0.40	0	1	1	0
123	2.76	-0.94	7.77	8.01	0.79	0	1.73	3.98	0	0	1	0
125	3.87	2.21	14.93	5.37	-2.14	0	5.12	2.66	0	1	0	0
126	4.13	1.87	21.40	33.38	0.15	0	5.44	0.88	0	0	0	1
127	3.98	0.99	15.24	39.60	0.10	0	2.49	0.11	0	0	0	1
129	4.35	1.25	5.61	9.46	-0.51	0	4.27	0.32	0	1	0	0
131	1.83	0.77	10.10	5.88	-2.35	0	6.15	2.30	0	1	0	0
132	3.650	1.58	9.86	20.69	-1.33	0	3.22	-1.70	0	1	1	0
133	1.11	0.61	19.26	22.24	-2.53	0	6.95	1.58	0	1	0	0
134	2.71	1.05	14.72	4.84	0.03	0	3.52	-1.05	0	1	1	0
137	5.68	2.59	23.50	33.00	-1.92	0	5.70	4.40	0	0	0	1
138	3.99	2.41	11.39	29.65	0.25	0	5.81	2.01	0	0	0	1
139	2.88	1.44	5.27	19.48	0.33	0	5.36	0.95	0	0	0	1
140	4.13	2.58	6.26	53.52	0.8	0	6.55	3.00	0	0	0	1
141	2.55	-0.88	15.24	33.51	-1.53	0	-2.29	2.50	0	1	1	0
143	3.61	2.02	8.81	25.71	-3.34	0	4.05	3.43	0	1	0	0
144	5.71	2.31	24.34	27.97	-0.75	0	4.76	-2.99	0	1	1	0
145	3.32	0.56	3.18	35.22	0.70	0	3.23	1.08	0	0	0	1
146	0.19	-0.14	10.55	70.54	0.78	0	-2.23	3.64	1	0	1	0
148	5.54	2.88	11.19	8.74	0.50	1	7.21	1.07	0	0	0	1
150	3.55	1.15	7.09	23.8	0.62	1	3.82	-2.35	0	0	1	0
151	1.44	-0.51	5.05	14.81	-5.60	0	-2.31	-0.41	0	1	1	0
152	3.49	-2.19	10.32	48.1	0.58	-1	-3.75	-2.99	0	0	1	0
153	0.16	-0.64	2.57	20.90	0.59	1	-0.80	0.31	1	0	0	0
156	3.63	2.13	11.98	30.57	-4.91	1	6.82	-4.49	0	1	1	0
157	3.8	-0.31	6.79	69.73	0.57	0	1.32	0.64	0	0	1	0
159	3.86	-0.40	20.42	46.67	-1.08	0	1.68	4.68	0	1	1	0
162	5.57	2.32	16.70	27.10	0.32	0	5.48	2.97	0	0	0	1
164	3.25	1.47	27.95	6.99	-1.07	0	3.41	0.40	0	1	0	0
166	3.92	1.37	12.46	38.77	-1.91	1	4.67	2.86	0	1	0	0
167	1.35	-0.50	16.98	30.62	-4.85	1	-3.88	2.45	1	1	0	0
170	3.65	1.35	7.76	27.60	0.69	0	3.2	2.42	0	0	0	1
171	3.63	1.81	32.54	12.20	-0.20	-1	4.34	2.50	0	1	0	0
172	3.83	0.42	14.14	37.10	-0.15	-1	3.56	3.33	0	1	0	0
174	3.65	2.26	5.16	19.64	0.65	0	5.66	-0.95	0	0	1	0
175	3.78	1.61	7.53	20.68	0.70	0	3.81	-4.26	0	0	1	0
176	6.18	1.35	10.77	50.24	0.76	0	4.32	3.30	0	0	0	1
180	4.77	2.01	12.99	5.77	-1.57	0	4.66	1.03	0	1	0	0

No.	R1	R2	R3	R4	R5	R6	R7	R8	Prof	STL	Debt	FA
181	4.95	1.80	16.06	46.24	-1.30	1	3.75	2.82	0	1	0	0
182	3.15	2.20	12.61	27.34	0.71	0	6.48	4.76	0	0	1	0
185	0.62	-1.05	27.58	6.15	-1.51	0	-3.21	-0.42	1	1	0	0
186	3.53	1.11	5.98	36.05	0.76	1	3.14	1.59	0	0	0	1
187	3.30	2.35	6.13	30.54	0.76	1	4.73	0.23	0	0	0	1
190	3.82	1.59	9.46	25.38	-0.92	0	5.86	2.61	0	1	0	0
191	5.28	2.17	18.43	28.64	0.12	0	6.64	4.67	0	0	1	0
193	2.44	1.24	20.63	29.84	-0.55	-1	3.80	-0.49	0	1	1	0
194	3.54	2.40	10.9	14.08	0.62	0	4.83	-5.82	0	0	1	0
195	2.66	1.34	12.31	10.80	0.17	0	3.70	-2.64	0	0	1	0
196	3.48	2.36	10.13	34.59	0.11	0	5.48	2.79	0	0	0	1
197	1.73	1.02	13.05	34.49	-5.28	1	2.16	-5.10	0	1	1	0
198	3.25	0.24	26.71	32.45	-3.01	1	0.85	-2.56	0	1	1	0
201	0.54	-0.68	8.22	430	0.20	-1	-1.25	3.50	1	0	0	0
204	0.84	-0.73	8.60	35.57	0.40	0	1.74	2.62	1	0	1	0
205	0.95	-0.31	9.84	19.61	0.40	-1	-1.79	1.35	1	0	0	0
206	0.72	0.39	7.82	15.53	-2.61	0	1.79	1.54	1	0	0	0
207	1.79	0.67	16.34	22.41	0.11	0	2.25	2.97	0	0	0	1
208	2.01	0.83	17.43	420	0.61	0	2.54	1.05	0	0	0	1
209	0.13	-1.91	170	55.42	0.63	1	-2.98	1.76	1	0	0	0
210	1.25	-2.09	19.73	25.14	0.56	0	1.86	4.25	1	0	1	0
212	0.45	-0.14	8.90	26.00	0.71	0	-1.29	2.26	1	0	0	0
213	0.87	0.13	14.28	21.20	0.03	0	4.69	3.64	1	0	0	0
214	0.27	0.19	31.78	24.34	0.25	-1	1.03	0.67	1	1	0	0
216	0.92	0.34	19.33	28.57	0.19	0	2.60	2.82	1	0	0	0
217	0.68	0.23	27.93	8.96	-1.32	0	4.11	1.95	1	1	0	0
219	0.25	0.11	19.38	18.90	0.32	0	2.82	0.45	1	0	0	0
222	0.56	0.13	13.91	26.81	0.47	0	4.25	2.48	1	0	0	0
223	1.14	-0.64	16.82	43.46	0.30	0	1.41	1.09	1	0	0	0
225	1.43	-0.30	12.25	26.67	0.39	1	-1.22	0.66	1	0	0	0
229	1.05	0.11	17.65	21.77	0.40	1	1.96	3.13	1	0	0	0
231	0.34	0.13	14.39	12.75	-1.62	1	2.47	1.33	1	1	0	0
232	1.26	0.73	32.00	18.95	-0.52	0	2.48	2.68	1	1	0	0
234	1.23	0.39	23.33	24.38	-1.25	-1	1.78	1.70	1	1	0	0
235	1.34	0.09	21.42	12.91	-3.03	-1	3.75	3.16	1	1	0	0
237	1.47	-1.60	32.10	10.92	-2.24	1	1.72	1.67	1	1	0	0
239	1.23	-2.03	26.76	7.61	-0.43	0	2.13	1.13	1	1	0	0
240	1.06	-1.36	24.21	5.13	-2.95	1	2.10	2.14	1	1	0	0
241	1.44	-0.77	15.89	27.61	0.49	0	1.76	-2.91	1	0	1	0
242	1.75	0.16	14.84	31.58	0.32	0	1.18	2.67	1	0	1	0
243	1.25	0.18	14.81	42.83	-0.16	1	0.98	-1.79	1	0	1	0
245	1.12	0.15	17.70	51.89	0.08	0	1.66	-2.40	1	0	1	0
247	0.39	0.27	6.67	44.08	0.74	-1	1.34	1.87	1	0	1	0
248	0.78	0.30	7.95	29.24	0.04	0	0.75	2.65	0	0	1	0
250	1.07	-0.15	9.84	19.08	-0.15	0	1.12	3.96	1	0	1	0
252	1.39	0.22	16.36	18.91	-0.16	1	1.35	3.80	1	0	1	0
253	0.64	0.15	14.09	33.03	0.20	0	1.19	3.93	1	0	1	0
255	1.25	0.21	8.63	25.05	-1.28	-1	6.02	2.45	0	0	0	1
256	0.73	0.18	14.58	30.30	0.32	0	1.61	3.88	1	0	1	0
257	0.52	-2.10	23.18	53.66	0.12	0	0.94	3.75	1	0	1	0
258	1.40	-1.27	16.90	47.40	0.15	1	-1.64	-2.40	1	0	1	0
260	0.30	0.20	6.69	40.42	0.39	0	1.83	4.24	1	0	1	0
264	1.27	0.12	17.92	20.73	-0.10	1	4.17	3.04	1	0	0	0
267	1.50	-0.66	19.33	25.56	0.48	1	2.22	0.96	1	0	0	0
268	1.56	0.06	19.70	18.67	0.33	0	5.08	2.12	1	0	0	0
269	1.23	0.15	19.67	17.70	0.31	0	3.75	1.94	1	0	0	0
270	0.30	0.08	12.72	25.34	0.03	1	2.97	1.08	1	0	0	0
272	1.22	0.33	16.71	15.98	0.28	0	1.14	2.90	1	0	1	0
273	1.78	-0.44	15.01	34.73	0.04	1	1.07	-2.24	1	0	1	0
277	0.21	-1.10	17.33	18.70	0.27	0	0.93	3.68	1	0	1	0
280	1.29	0.25	11.49	16.86	0.11	0	3.28	-4.20	1	0	1	0
283	0.35	-0.70	27.10	5.60	-0.37	1	1.89	2.33	1	1	0	0
284	-0.06	-0.71	28.42	32.24	-4.52	-1	-1.46	0.42	1	1	0	0

No.	R1	R2	R3	R4	R5	R6	R7	R8	Prof	STL	Debt	FA
285	1.11	0.06	30.84	11.94	-4.67	1	-0.36	0.46	1	1	0	0
286	1.12	-0.86	32.22	14.08	0.14	-1	2.01	3.12	1	1	0	0
289	0.68	-1.42	27.65	5.01	-0.20	-1	2.18	0.64	1	1	0	0
290	0.87	0.15	10.05	54.20	0.68	1	4.40	3.63	1	0	0	0
291	0.68	-0.24	6.05	52.46	0.67	1	2.26	0.39	1	0	0	0
292	0.36	-0.14	7.54	49.49	0.30	1	2.16	0.45	1	0	0	0
294	1.37	0.21	5.33	49.80	0.38	0	4.52	1.67	1	0	0	0
295	1.02	-0.83	6.56	57.86	0.79	0	2.16	1.18	1	0	0	0
297	0.30	0.15	7.36	51.32	0.43	0	4.38	1.30	1	0	0	0
298	0.70	0.11	6.95	62.82	0.38	1	6.40	1.22	1	0	0	0
299	1.36	-0.09	15.50	33.89	0.10	0	2.01	-1.23	1	0	1	0
300	1.06	-1.14	12.80	36.62	0.30	1	2.04	4.98	1	0	1	0
301	1.22	0.23	17.80	28.90	0.10	1	1.87	4.36	1	0	1	0
305	1.50	0.012	24.70	31.62	0.21	0	3.01	-2.05	1	0	1	0

7.3. Appendix C:

Samples Used to Test the Neural Networks

No.	R1	R2	R3	R4	R5	R6	R7	R8	Prof	STL	Debt	FA
5	1.51	-1.49	2.67	11.41	0.21	0	-2.17	1.05	1	1	0	0
8	2.20	0.80	9.88	19.09	0.13	-1	2.45	-2.97	0	1	1	0
9	2.01	0.86	17.43	46.75	0.61	0	3.39	0.21	0	0	0	1
17	3.62	1.80	23.74	-5.00	0.30	0	4.60	4.54	0	0	0	1
18	4.02	1.70	13.21	20.88	0.70	1	4.30	-1.45	0	0	1	0
20	6.05	1.70	14.22	35.48	-0.07	-1	4.26	1.14	0	1	0	0
28	6.18	2.31	4.24	11.26	0.46	1	5.60	2.48	0	0	0	1
32	2.58	0.90	11.46	24.26	-3.47	0	2.20	4.84	0	1	1	0
33	2.05	0.80	10.02	22.52	-0.84	0	2.24	-1.91	0	1	0	0
35	2.85	-2.29	23.62	21.43	0.58	0	-2.72	-3.02	0	0	1	0
40	3.01	0.26	9.70	68.10	0.72	0	5.76	1.06	0	0	0	1
41	1.97	-0.32	9.10	77.62	0.89	0	1.97	-0.58	0	0	1	0
49	5.23	2.23	20.23	27.39	-0.33	-1	5.46	-4.79	0	1	1	0
51	4.13	1.76	17.30	43.35	-3.77	0	4.30	0.75	0	1	0	0
52	2.54	0.34	20.03	20.42	-1.37	0	3.92	-1.54	0	1	1	0
53	4.10	2.32	8.15	57.76	0.23	0	6.26	0.78	0	0	0	1
57	3.42	1.88	29.76	18.15	0.56	-1	5.22	4.10	0	1	0	0
60	2.30	-1.35	18.14	5.15	-2.61	0	-3.48	4.80	0	1	1	0
62	2.54	0.88	19.10	-6.55	0.12	1	3.12	-3.15	0	1	1	0
68	4.21	2.27	20.46	9.26	-1.58	0	5.73	1.54	0	1	0	0
83	1.30	0.72	22.15	20.48	0.79	0	4.07	1.24	1	0	0	0
84	2.74	1.21	7.39	40.98	0.63	0	5.22	-2.87	0	0	1	0
86	2.25	-2.08	16.57	34.26	0.09	0	-2.31	-0.99	0	0	1	0
87	0.33	-0.14	33.20	6.28	-0.47	1	-3.58	0.02	1	1	0	0
92	5.45	2.24	2.63	83.90	0.70	0	6.23	2.98	0	0	0	1
95	1.20	-1.25	22.63	43.86	-3.97	0	-2.12	0.80	1	1	0	0
96	4.70	2.59	5.84	53.04	0.37	1	6.54	-3.26	0	0	1	0
102	2.23	-1.46	21.59	41.94	-5.15	0	-1.67	-3.12	0	1	1	0
104	6.15	2.98	10.38	17.45	0.36	0	7.20	-4.87	0	0	1	0
105	4.35	1.72	18.76	10.71	-0.06	-1	4.60	1.97	0	1	0	0
108	4.55	0.69	23.66	12.81	-3.29	1	7.70	2.16	0	1	0	0
112	3.59	1.63	17.30	4.56	-4.86	0	7.58	-3.71	0	1	1	0
114	-0.4	-1.23	15.12	38.03	-4.66	0	-3.59	-3.11	1	0	1	0
115	3.12	-0.76	11.21	13.83	-2.4	0	-2.35	0.77	0	1	1	0
121	5.13	0.36	15.36	55.18	0.15	0	2.91	0.25	0	0	0	1
122	4.18	2.32	11.99	7.77	-0.45	0	4.71	2.90	0	1	0	0
124	4.04	-0.21	7.63	20.43	0.73	0	1.54	-4.59	0	0	1	0
128	3.47	-1.35	13.01	12.04	-1.17	0	2.04	2.17	0	1	1	0
130	2.12	-1.63	20.38	-2.79	0.23	1	-2.37	-0.20	0	0	1	0
135	4.94	1.37	9.80	21.15	0.29	0	3.58	-3.42	0	0	1	0
136	3.47	0.97	8.97	28.73	0.09	0	3.37	3.87	0	0	1	0
142	3.80	1.42	18.12	36.76	0.67	1	5.93	-1.73	0	0	1	0
147	0.21	-2.25	4.42	24.11	0.12	1	-2.86	3.49	1	0	0	0
149	4.90	-1.11	14.72	26.05	0.28	0	2.26	4.12	0	0	1	0
154	-0.12	-1.39	16.29	31.62	-2.56	0	-2.22	3.25	1	1	0	0
155	1.23	0.53	9.82	20.41	0.25	0	2.62	-1.71	1	0	1	0
158	3.85	2.12	30.77	59.17	0.75	1	5.61	1.08	0	0	0	1
160	3.17	1.67	18.78	2.19	-1.33	0	3.46	-1.27	0	1	1	0
161	1.35	-1.55	17.24	33.69	-0.03	1	-2.45	1.2	1	1	0	0
163	1.55	-0.97	36.48	6.28	-0.60	-1	2.29	0.16	1	1	0	0
165	4.25	2.21	14.11	9.98	0.22	0	5.31	0.52	0	0	0	1
168	4.29	1.29	25.53	14.52	-3.97	0	3.46	1.47	0	1	0	0
169	0.63	-0.17	27.30	11.58	-1.09	0	-0.77	0.26	1	1	0	0
173	3.58	0.51	19.39	19.93	-0.31	0	4.27	2.66	0	1	0	0
177	1.39	-0.36	12.46	32.99	-1.28	0	-2.49	1.78	1	1	0	0
178	3.89	2.21	14.95	48.91	0.55	1	5.7	0.73	0	0	0	1
179	4.67	3.10	9.61	27.03	-1.83	-1	7.2	2.12	0	1	0	0
183	2.18	1.38	15.64	46.84	-1.49	0	3.45	4.81	0	1	1	0
184	-0.41	-1.17	26.24	12.01	-0.41	-1	-2.21	0.93	1	1	0	0

No.	R1	R2	R3	R4	R5	R6	R7	R8	Prof	STL	Debt	FA
188	2.85	1.73	20.76	17.85	-2.34	-1	3.75	1.26	0	1	0	0
189	1.34	-1.09	6.39	42.01	-0.27	-1	-1.65	0.89	1	1	0	0
192	3.58	2.13	16.40	9.21	-0.43	-1	5.06	4.26	0	1	1	0
199	2.90	1.09	10.09	28.37	0.15	0	3.29	-4.74	0	0	1	0
200	4.23	2.03	15.92	25.91	-0.57	0	4.75	4.63	0	1	1	0
202	0.67	0.43	27.45	10.00	-1.39	0	4.91	0.67	1	1	0	0
203	1.16	0.41	13.87	44.27	0.05	-1	4.64	1.54	1	0	0	0
211	0.60	0.44	20.66	35.30	0.62	0	1.40	2.40	1	0	0	0
215	1.02	-1.33	23.74	35.00	0.30	0	-2.11	0.37	1	0	0	0
218	1.22	0.25	16.71	21.72	0.18	1	3.80	2.05	1	0	0	0
220	-1.01	-2.32	12.02	23.88	0.27	0	-2.69	1.41	1	0	0	0
221	0.54	0.25	17.41	22.05	-0.10	1	1.67	0.04	1	0	0	0
224	0.37	0.19	16.03	31.18	0.53	0	2.23	2.30	1	0	0	0
226	0.75	-0.41	16.44	25.98	0.17	0	0.70	1.22	1	0	0	0
227	1.35	-0.4	12.35	22.78	0.57	0	1.78	2.54	1	0	0	0
228	0.67	0.38	18.26	31.82	-0.20	1	2.28	3.01	1	0	0	0
230	1.27	-1.16	14.82	27.03	0.52	0	2.30	2.43	1	0	0	0
233	0.09	-1.06	12.16	33.88	-2.89	-1	-1.36	0.72	1	1	0	0
236	1.13	0.57	20.22	6.80	0.21	-1	2.56	0.58	1	1	0	0
238	1.68	-1.88	30.39	15.3	0.10	0	1.77	2.30	1	1	0	0
244	0.23	-1.29	12.95	14.03	0.33	0	0.43	4.12	1	0	1	0
246	0.86	-1.19	8.49	55.97	0.22	0	1.15	1.77	1	0	1	0
249	0.47	-0.32	13.77	33.17	-0.20	0	1.28	4.18	1	0	1	0
251	0.85	0.2	13.88	15.77	0.32	0	1.2	3.28	1	0	1	0
254	0.82	-0.29	15.21	23.64	0.28	-1	-0.42	-0.89	1	0	1	0
259	0.47	-0.08	14.66	22.30	0.08	0	-0.43	-1.75	1	0	1	0
261	1.31	0.15	15.78	38.12	0.05	1	1.20	4.45	1	0	1	0
262	1.28	0.13	16.73	45.78	-0.10	1	1.27	-0.33	1	0	1	0
263	1.16	0.32	19.04	25.56	0.47	-1	1.66	3.89	1	0	1	0
265	0.80	-0.14	13.68	29.10	0.16	0	2.30	0.71	1	0	0	0
266	1.13	0.10	23.20	22.66	0.20	0	3.57	2.44	1	0	0	0
271	1.14	0.34	20.44	14.62	0.62	0	3.87	2.47	1	0	0	0
274	1.24	0.11	7.39	40.98	0.63	0	1.48	1.43	1	0	1	0
275	1.28	0.25	21.76	38.67	0.43	0	1.04	3.68	1	0	1	0
276	0.34	-0.86	12.82	28.87	0.47	1	0.81	4.36	1	0	1	0
278	0.11	-1.40	12.38	25.42	0.03	1	1.41	-3.01	1	0	1	0
279	1.32	0.11	12.13	24.15	0.46	0	1.16	3.35	1	0	1	0
281	3.52	2.24	2.63	83.90	-0.70	0	5.54	3.73	0	0	0	1
282	1.36	0.30	29.48	22.25	0.14	0	3.38	2.21	1	1	0	0
287	0.32	0.22	31.10	26.90	0.03	-1	2.63	0.45	1	1	0	0
288	1.17	0.08	21.81	4.22	-0.18	0	3.84	0.65	1	1	0	0
293	1.02	0.26	9.23	50.84	0.60	0	3.50	0.40	1	0	0	0
296	0.55	0.24	6.30	66.68	0.22	0	1.99	0.45	1	0	0	0
302	1.43	0.24	19.70	24.60	-0.10	1	1.14	4.70	1	0	1	0
303	1.22	0.12	21.30	15.40	0.05	1	1.44	-2.03	1	0	1	0
304	1.25	0.30	18.00	23.40	0.15	1	2.45	-3.4	1	0	1	0

7.4. Appendix D:

Results of the Backpropogation Training

a) Scaling Interval: [0,1]									
Date	No. Hidden Neurons	Activation Hidden	Function Output	Training				Testing	
				Time	*Epochs*	*RMS*	*% Error*	*RMS*	*% Error*
21-Aug-95	10	logsig	logsig	3572	37	0.0938	3	0.3166	53
20-Aug-95	10	logsig	logsig	5525	40	0.1939	15	0.2328	38
21-Aug-95	10	logsig	logsig	3501	40	0.1044	7	0.2328	38
20-Aug-95	15	logsig	logsig	13835	50	0.1490	20	0.3378	60
21-Aug-95	15	logsig	logsig	11662	52	0.0794	1	0.2757	49
20-Aug-95	20	logsig	logsig	21081	50	0.2776	51	0.3582	71
20-Aug-95	25	logsig	logsig	24810	50	0.0546	1	0.2973	54
20-Aug-95	30	logsig	logsig	45877	50	0.0511	0	0.3064	54
20-Aug-95	40	logsig	logsig	66107	50	0.1883	36	0.3053	58
11-Aug-95	10	logsig	tansig	7624	150	0.1496	23	0.2935	47
21-Aug-95	10	logsig	tansig	5878	70	0.1598	23	0.2735	43
20-Aug-95	15	logsig	tansig	12879	50	0.1231	13	0.3177	63
22-Aug-95	15	logsig	tansig	13748	70	0.1250	12	0.3199	55
11-Aug-95	20	logsig	tansig	15786	150	0.1033	6	0.3322	68
20-Aug-95	25	logsig	tansig	43678	50	0.0802	0	0.3777	71
16-Aug-95	30	logsig	tansig	101726	100	0.0586	0	0.4830	90
21-Aug-95	40	logsig	tansig	132520	50	0.0450	0	0.5011	90
14-Aug-95	10	tansig	logsig	12897	103	0.0886	0	0.3400	46
21-Aug-95	10	tansig	logsig	5777	70	0.0908	0	0.3401	49
20-Aug-95	15	tansig	logsig	11004	41	0.0653	0	0.3025	48
21-Aug-95	15	tansig	logsig	9798	41	0.0653	0	0.3025	48
14-Aug-95	20	tansig	logsig	20168	45	0.0549	0	0.3076	52
20-Aug-95	25	tansig	logsig	23524	50	0.1055	8	0.2904	49
15-Aug-95	30	tansig	logsig	54022	50	0.0336	0	0.3326	58
20-Aug-95	40	tansig	logsig	102635	50	0.2307	42	0.3343	68
20-Aug-95	10	tansig	tansig	4608	34	0.1473	20	0.2600	42
21-Aug-95	10	tansig	tansig	3360	34	0.1473	20	0.2600	42
20-Aug-95	15	tansig	tansig	12031	50	0.1199	8	0.3183	62
22-Aug-95	15	tansig	tansig	13626	70	0.1177	10	0.3200	68
20-Aug-95	20	tansig	tansig	18569	50	0.0907	6	0.3721	62
20-Aug-95	25	tansig	tansig	31624	50	0.1046	5	0.3783	81
20-Aug-95	30	tansig	tansig	34833	50	0.0814	2	0.4467	78
21-Aug-95	40	tansig	tansig	101526	50	0.2008	47	0.4976	91

b) Scaling Interval: [-1,1]									
Date	**Hidden Neurons**	**Activation Function**		**Training**				**Testing**	
		Hidden	**Output**	*Time*	*Epochs*	*RMS*	*% Error*	*RMS*	*% Error*
20-Aug-95	10	logsig	logsig	6956	50	0.1575	16	0.2743	41
21-Aug-95	10	logsig	logsig	7525	70	0.1019	3	0.3124	42
21-Aug-95	10	logsig	logsig	3235	74	0.1231	9	0.3182	44
20-Aug-95	15	logsig	logsig	8765	50	0.0726	0	0.3055	55
21-Aug-95	15	logsig	logsig	7783	37	0.0726	0	0.3055	55
20-Aug-95	20	logsig	logsig	18978	50	0.2378	51	0.3103	64
22-Aug-95	20	logsig	logsig	8727	70	0.1255	14	0.2745	49
20-Aug-95	25	logsig	logsig	28925	50	0.1565	28	0.2895	54
20-Aug-95	30	logsig	logsig	14158	50	0.2118	43	0.3193	59
20-Aug-95	40	logsig	logsig	46334	50	0.1892	33	0.3298	54
12-Aug-95	10	logsig	tansig	1415	53	0.1057	5	0.2803	41
21-Aug-95	10	logsig	tansig	7538	70	0.1607	16	0.2415	44
11-Aug-95	10	logsig	tansig	2005	100	0.1527	17	0.2482	43
20-Aug-95	15	logsig	tansig	10031	50	0.1322	11	0.2674	51
21-Aug-95	15	logsig	tansig	14239	70	0.1244	10	0.2832	54
22-Aug-95	20	logsig	tansig	8698	50	0.0963	5	0.3821	51
13-Aug-95	20	logsig	tansig	30588	96	0.0668	3	0.3787	51
20-Aug-95	25	logsig	tansig	18667	50	0.0991	7	0.3827	62
13-Aug-95	30	logsig	tansig	85013	130	0.0482	0	0.5305	70
20-Aug-95	40	logsig	tansig	53041	50	0.1577	15	0.5731	79
22-Aug-95	40	logsig	tansig	8727	70	0.1255	14	0.2745	49
13-Aug-95	10	tansig	logsig	4279	57	0.0795	2	0.3110	40
21-Aug-95	10	tansig	logsig	6595	57	0.0795	2	0.3110	40
20-Aug-95	15	tansig	logsig	18022	50	0.1211	9	0.2971	48
21-Aug-95	15	tansig	logsig	14187	70	0.0689	0	0.3109	49
22-Aug-95	20	tansig	logsig	8220	70	0.0714	1	0.3069	52
13-Aug-95	20	tansig	logsig	15796	71	0.0683	0	0.3059	53
20-Aug-95	25	tansig	logsig	49060	50	0.2325	45	0.2840	56
13-Aug-95	30	tansig	logsig	50212	150	0.0710	3	0.2920	55
14-Aug-95	40	tansig	logsig	116776	138	0.0266	0	0.3156	53
20-Aug-95	10	tansig	tansig	8836	50	0.1523	18	0.2716	53
21-Aug-95	10	tansig	tansig	7289	70	0.1504	17	0.2803	53
20-Aug-95	15	tansig	tansig	8589	50	0.1341	11	0.3393	65
21-Aug-95	15	tansig	tansig	14314	70	0.1302	11	0.3406	64
20-Aug-95	20	tansig	tansig	26743	50	0.1273	10	0.3588	56
22-Aug-95	20	tansig	tansig	9727	70	0.1042	7	0.4176	55
20-Aug-95	25	tansig	tansig	49467	50	0.1327	13	0.3673	75
20-Aug-95	30	tansig	tansig	32702	50	0.2301	28	0.5048	76
21-Aug-95	40	tansig	tansig	130381	50	0.2171	31	0.5061	81
5-Sep-95	40	tansig	tansig	38736	70	0.1379	13	0.5887	81

c) Scaling Intervals: [0,1] (R1), [0,1] (R2), [-1,1] (R3), [-1,1] (R4), [0,1] (R5), [-1,1] (R6), [0,1] (R7), [-1,1] (R8)									
Date	**Hidden Neurons**	**Activation Function**		**Training**				**Testing**	
		Hidden	**Output**	*Time*	*Epochs*	*RMS*	*% Error*	*RMS*	*% Error*
23-Aug-95	10	logsig	logsig	2306	50	0.1381	16	0.2439	38
23-Aug-95	10	logsig	logsig	2770	70	0.1077	6	0.3041	40
23-Aug-95	15	logsig	logsig	7666	50	0.0869	1	0.2833	47
23-Aug-95	20	logsig	logsig	15354	50	0.0796	3	0.3026	47
23-Aug-95	25	logsig	logsig	22898	50	0.1750	12	0.3082	61
23-Aug-95	30	logsig	logsig	19546	50	0.1807	29	0.3187	56
28-Aug-95	40	logsig	logsig	26609	50	0.0906	4	0.3361	70
22-Aug-95	10	logsig	tansig	1633	50	0.1560	20	0.2825	50
22-Aug-95	10	logsig	tansig	1107	51	0.1560	20	0.2853	50
23-Aug-95	15	logsig	tansig	10455	50	0.1378	17	0.3374	61
28-Aug-95	20	logsig	tansig	7659	50	0.0957	7	0.3912	61
28-Aug-95	25	logsig	tansig	12238	50	0.1587	25	0.3599	73
28-Aug-95	30	logsig	tansig	32918	130	0.0651	2	0.5154	83
23-Aug-95	10	tansig	logsig	3509	50	0.0905	5	0.2773	41
23-Aug-95	10	tansig	logsig	6151	70	0.0906	2	0.3396	44
28-Aug-95	15	tansig	logsig	5499	50	0.1529	21	0.2703	51
28-Aug-95	20	tansig	logsig	11022	50	0.0832	2	0.3104	60
28-Aug-95	30	tansig	logsig	16399	70	0.0415	0	0.3026	51
23-Aug-95	10	tansig	tansig	1119	50	0.1577	21	0.2620	50
23-Aug-95	10	tansig	tansig	1652	70	0.1512	20	0.3279	53
28-Aug-95	15	tansig	tansig	4220	50	0.1211	12	0.3037	59
28-Aug-95	20	tansig	tansig	5863	50	0.1356	20	0.3442	68
29-Aug-95	30	tansig	tansig	14474	50	0.2172	36	0.4365	88

7.5. Appendix E:

Results of the RBF Training

3 Networks
K-Means (with P = 10 and Normalized Gaussian) (expected number of centers = 100)

Date	Scaling	λ	Orth. Accuracy	Final Centers			Training			Testing	
							Time	*RMS*	*% Error*	*RMS*	*% Error*
12-Sep-95	30-30%	0	0.2	100	100	100	8061	0.1753	30	0.2990	50
18-Aug-95	30-30%	0	0.3	34	43	46	6774	0.2371	42	0.3436	67
18-Aug-95	30-30%	0	0.4	26	34	37	6088	0.2624	44	0.3509	66
18-Aug-95	30-30%	0	0.5	18	26	28	5368	0.2911	52	0.3606	79
19-Aug-95	30-30%	1	0.2	100	100	100	13849	0.2239	54	0.3129	63
19-Aug-95	30-30%	2	0.2	100	100	100	13803	0.3114	73	0.3570	78
19-Aug-95	40-40%	0	0.2	100	100	100	15509	0.1753	30	0.2990	50
11-Sep-95	40-40%	0	0.3	34	43	46	4200	0.2336	46	0.3437	64
11-Sep-95	40-40%	0	0.4	29	35	32	4555	0.2761	53	0.3642	71
11-Sep-95	40-40%	0	0.5	22	27	23	2175	0.2929	57	0.3491	71
19-Aug-95	50-50%	0	0.2	100	100	100	14584	0.1847	32	0.3105	52

3 Networks
K-Means (with P = 5 and Normalized Gaussian) (expected number of centers = 100)

Date	Scaling	λ	Orth. Accuracy	Final Centers			Training			Testing	
							Time	*RMS*	*% Error*	*RMS*	*% Error*
22-Aug-95	30-30%	0	0.2	100	100	100	10323	0.1508	23	0.2827	40
22-Aug-95	30-30%	0	0.3	24	34	37	4442	0.2609	49	0.3324	72
22-Aug-95	30-30%	0	0.4	17	26	27	2147	0.2789	54	0.3412	73
22-Aug-95	30-30%	0	0.5	13	19	19	2905	0.3125	67	0.3651	81
22-Aug-95	30-30%	1	0.1	100	100	100	8609	0.2276	56	0.2966	56
22-Aug-95	30-30%	1	0.3	24	34	37	2470	0.3744	72	0.3865	71
22-Aug-95	30-30%	2	0.1	100	100	100	4978	0.2976	72	0.3093	72
11-Sep-95	40-40%	0	0.2	100	100	100	10666	0.1645	28	0.2795	48
11-Sep-95	40-40%	0	0.3	29	37	33	4224	0.2707	53	0.3527	74
11-Sep-95	40-40%	0	0.4	21	28	24	4163	0.2892	55	0.3513	77
11-Sep-95	50-50%	0	0.2	100	100	100	12236	0.1702	31	0.2961	52

3 Networks
Normalized Gaussian (expected number of centers = 100)

Date	Center Det.	l	Orth. Accuracy	Final Centers			Training			Testing	
							Time	*RMS*	*% Error*	*RMS*	*% Error*
scaling: 80-80; 80-80; 50-50; 80-80; 0-0; 80-80; 80-80											
18-Aug-95	First M	0	0.1	22	40	100	6712	0.1760	33	0.2453	53
18-Aug-95	First M	0	0.1	22	40	100	6676	0.1879	31	0.2324	50
scaling: 60-60; 60-60; 60-60; 60-60; 0-0; 60-60; 60-60											
12-Sep-95	First M	0	0.1	19	33	100	1888	0.1698	29	0.2470	45
12-Sep-95	K-M(10)	0	0.1	100	100	100	3177	0.1872	29	0.3378	52

3 Networks
K-Means (with P = 20 and Normalized Gaussian) (expected number of centers = 200)

Date	Scaling	λ	Orth. Accuracy	Final centers			Training Time	Training RMS	Training % Error	Testing RMS	Testing % Error
22-Aug-95	30-30%	0	0.01	141	129	128	54737	0.1388	25	0.3242	50
05-Sep-95	30-30%	0	0.10	67	80	81	6726	0.2473	48	0.3572	68
21-Aug-95	30-30%	0	0.20	59	71	72	7241	0.2555	52	0.3602	70
06-Sep-95	30-30%	0	0.30	52	62	63	6525	0.2628	55	0.3449	73
06-Sep-95	30-30%	0	0.40	45	54	54	52482	0.2673	55	0.3443	75
22-Aug-95	30-30%	0	0.50	37	45	45	15885	0.2668	58	0.3487	77
09-Sep-95	30-30%	1	0.01	141	129	128	31814	0.3161	86	0.3649	88
21-Aug-95	30-30%	1	0.10	67	80	81	35470	0.3606	84	0.4042	86
06-Sep-95	30-30%	1	0.20	59	71	72	11784	0.3511	81	0.4042	85
21-Aug-95	30-30%	2	0.01	141	129	128	71565	0.4429	86	0.4480	89
21-Aug-95	30-30%	2	0.10	67	80	81	34932	0.4453	85	0.4576	88
21-Aug-95	30-30%	2	0.20	59	71	72	32974	0.4211	86	0.4479	88
20-Aug-95	40-40%	0	0.01	141	129	128	27818	0.1511	30	0.3133	51
20-Aug-95	40-40%	0	0.10	67	80	81	18380	0.2414	45	0.3587	66
20-Aug-95	40-40%	0	0.20	59	71	72	16107	0.2502	50	0.3608	70
20-Aug-95	50-50%	0	0.01	141	129	128	18919	0.1533	32	0.3095	57
20-Aug-95	50-50%	0	0.10	67	80	81	12615	0.2358	52	0.3483	72
20-Aug-95	50-50%	0	0.20	59	71	72	10715	0.2434	53	0.3517	72

3 Networks
K-Means (with P = 10 and Normalized Gaussian) (expected number of centers = 200)

Date	Scaling	λ	Orth. Accuracy	Final centers			Training Time	Training RMS	Training % Error	Testing RMS	Testing % Error
07-Sep-95	30-30%	0	0.01	141	129	128	63772	0.1443	28	0.2986	48
22-Aug-95	30-30%	0	0.10	67	80	81	5221	0.2349	49	0.3512	67
22-Aug-95	30-30%	0	0.20	59	71	72	5066	0.2411	54	0.3645	68
22-Aug-95	30-30%	0	0.30	52	62	63	13048	0.2511	56	0.3586	75
22-Aug-95	30-30%	0	0.40	45	54	54	11458	0.2588	55	0.3546	76
22-Aug-95	30-30%	0	0.50	37	45	45	5091	0.2668	58	0.3487	77
07-Sep-95	30-30%	1	0.01	141	129	128	31732	0.3335	87	0.3525	81
09-Sep-95	30-30%	1	0.10	67	80	81	21951	0.3676	84	0.3895	83
09-Sep-95	30-30%	1	0.20	59	71	72	5201	0.3627	84	0.3888	85
09-Sep-95	30-30%	2	0.01	141	129	128	22678	0.4436	86	0.4288	88
09-Sep-95	30-30%	2	0.10	67	80	81	5824	0.4389	84	0.4384	87
09-Sep-95	30-30%	2	0.20	59	71	72	8600	0.4174	84	0.4307	88
09-Sep-95	40-40%	2	0.01	141	129	128	31534	0.1515	29	0.3056	50
09-Sep-95	40-40%	2	0.10	67	80	81	7283	0.2324	48	0.3571	71
09-Sep-95	40-40%	2	0.20	59	71	72	15150	0.2411	50	0.3564	73
12-Sep-95	50-50%	0	0.01	141	129	128	21465	0.1413	27	0.3191	56
12-Sep-95	50-50%	0	0.10	67	80	81	12867	0.2251	50	0.3494	69
12-Sep-95	50-50%	0	0.20	59	71	72	10830	0.2306	51	0.3456	70
12-Sep-95	60-60%	0	0.01	141	129	128	15396	0.1472	30	0.3288	64
12-Sep-95	60-60%	0	0.10	67	80	81	11636	0.2123	43	0.3770	76
12-Sep-95	60-60%	0	0.20	59	71	72	4187	0.2227	47	0.3726	76

3 Networks
First 200 Candidates

Date	Scaling	λ	Orth. Accuracy	Final centers			Training			Testing	
							Time	*RMS*	*% Error*	*RMS*	*% Error*
15-Sep-95	30-30%	0	0.01	150	138	145	17626	0.0173	1	0.3311	58
18-Sep-95	30-30%	0	0.05	74	78	81	25353	0.1005	11	0.2619	42
14-Sep-95	30-30%	0	0.10	15	32	42	6164	0.1651	26	0.2341	40
14-Sep-95	30-30%	0	0.20	2	2	7	4697	0.2433	49	0.2359	53
15-Sep-95	40-40%	0	0.01	138	148	142	24696	0.0134	0	0.3309	60
18-Sep-95	40-40%	0	0.05	65	82	82	12484	0.0973	11	0.2818	48
14-Sep-95	40-40%	0	0.10	17	27	50	4945	0.1695	30	0.2481	43
14-Sep-95	40-40%	0	0.20	2	3	8	3460	0.2489	48	0.2466	52
14-Sep-95	50-50%	0	0.01	139	142	142	16547	0.0216	1	0.3371	62
14-Sep-95	50-50%	0	0.10	14	33	50	2561	0.1696	29	0.2486	49
14-Sep-95	50-50%	0	0.20	3	4	7	1646	0.2599	62	0.2689	70
13-Sep-95	60-60%	0	0.01	132	147	143	13434	0.0159	1	0.3256	57
15-Sep-95	60-60%	0	0.02	106	128	119	14515	0.0367	1	0.3192	57
15-Sep-95	60-60%	0	0.03	91	110	103	37034	0.0581	5	0.2919	49
13-Sep-95	60-60%	0	0.04	75	100	92	18776	0.0761	7	0.2790	51
13-Sep-95	60-60%	0	0.05	61	85	83	6741	0.0924	10	0.2806	52
13-Sep-95	60-60%	0	0.06	52	70	75	5343	0.1090	18	0.2741	47
13-Sep-95	60-60%	0	0.07	43	59	70	4206	0.1251	20	0.2600	45
13-Sep-95	60-60%	0	0.08	34	51	67	3062	0.1431	22	0.2487	45
14-Sep-95	60-60%	0	0.09	25	43	61	3100	0.1622	26	0.2422	43
13-Sep-95	60-60%	0	0.10	21	32	53	2626	0.1708	30	0.2471	44
13-Sep-95	60-60%	0	0.15	7	12	21	5238	0.2270	43	0.2409	51
13-Sep-95	60-60%	0	0.20	3	6	8	4855	0.2500	62	0.2468	64

4 Networks
K-Means (with P = 10 and Normalized Gaussian) (expected number of centers = 100)

Date	Scaling	λ	Orth. Accuracy	Final Centers				Training			Testing	
								Time	*RMS*	*% Error*	*RMS*	*% Error*
23-Aug-95	30-30%	0	0.2	100	100	100	100	3806	0.2053	33	0.2900	50
22-Aug-95	30-30%	0	0.3	34	43	46	100	2483	0.2557	53	0.3245	72
22-Aug-95	30-30%	0	0.4	26	34	37	25	3235	0.2736	56	0.3304	71
22-Aug-95	30-30%	0	0.5	18	26	28	13	919	0.2946	62	0.3380	84
14-Aug-95	30-30%	1	0.1	100	100	100	100	20232	0.2292	62	0.4583	89
14-Aug-95	30-30%	1	0.2	100	100	100	100	20270	0.2292	62	0.4583	89
14-Aug-95	30-30%	2	0.1	100	100	100	100	20302	0.2874	86	0.4632	91
14-Aug-95	30-30%	2	0.2	100	100	100	100	5053	0.2874	86	0.4632	91
10-Sep-95	40-40%	0	0.2	100	100	100	100	3660	0.2150	39	0.2895	55
09-Sep-95	40-40%	0	0.3	100	45	41	100	6774	0.2533	56	0.3246	70
09-Sep-95	40-40%	0	0.4	29	35	32	18	1748	0.2835	61	0.3410	76

4 Networks
K-Means (with P = 5 and Normalized Gaussian) (expected number of centers = 100)

Date	Scaling	λ	Orth. Accuracy	Final Centers				Training Time	Training RMS	Training % Error	Testing RMS	Testing % Error
22-Aug-95	30-30%	0	0.1	100	100	100	100	4013	0.2007	34	0.2769	47
22-Aug-95	30-30%	0	0.2	33	44	49	100	2506	0.2520	50	0.3227	79
22-Aug-95	30-30%	0	0.3	24	34	37	100	2262	0.2725	60	0.3157	77
22-Aug-95	30-30%	0	0.4	17	26	27	15	1349	0.2856	65	0.3226	80
22-Aug-95	30-30%	0	0.5	13	19	19	10	3834	0.3106	78	0.3417	87
22-Aug-95	30-30%	1	0.1	100	100	100	100	4078	0.2491	68	0.2877	62
22-Aug-95	30-30%	1	0.3	24	34	37	100	3515	0.3582	82	0.3589	79
22-Aug-95	30-30%	2	0.1	100	100	100	100	3194	0.2994	84	0.2975	80
22-Aug-95	30-30%	2	0.3	24	34	37	100	2439	0.3536	76	0.3728	79
9-Sep-95	40-40%	0	0.1	100	100	100	100	3879	0.2086	38	0.2745	53
9-Sep-95	40-40%	0	0.3	29	37	33	66	2128	0.2796	63	0.3318	79
9-Sep-95	40-40%	0	0.4	21	28	24	16	1222	0.2931	67	0.3307	82
12-Sep-95	50-50%	0	0.2	100	100	100	100	12257	0.2475	50	0.3086	66

4 Networks
K-Means (with P = 20 and Normalized Gaussian) (expected number of centers = 200)

RBF Tests

Date	Scaling	λ	Orth. Accuracy	Final centers				Training Time	Training RMS	Training % Error	Testing RMS	Testing % Error
19-Aug-95	30-30%	0	0.01	141	129	128	175	33858	0.1940	37	0.3092	57
24-Aug-95	30-30%	0	0.10	67	80	81	28	8315	0.2628	59	0.3354	76
23-Aug-95	30-30%	0	0.20	59	71	72	25	11391	0.2686	62	0.3377	78
23-Aug-95	30-30%	0	0.30	52	62	63	22	5435	0.2739	65	0.3255	81
24-Aug-95	30-30%	0	0.40	45	54	54	19	13210	0.2771	66	0.3251	83
10-Sep-95	30-30%	0	0.50	37	45	45	16	12478	0.2823	73	0.2823	85
19-Aug-95	30-30%	1	0.01	141	129	128	175	34619	0.3133	100	0.3415	95
05-Sep-95	30-30%	1	0.10	67	80	81	28	10851	0.3474	98	0.3732	96
24-Aug-95	30-30%	1	0.20	59	71	72	25	16604	0.3401	94	0.3732	95
19-Aug-95	30-30%	2	0.01	141	129	128	175	47827	0.4127	100	0.4090	99
07-Sep-95	30-30%	2	0.10	67	80	81	28	10073	0.4146	100	0.4169	98
24-Aug-95	30-30%	2	0.20	67	80	81	28	10484	0.3952	100	0.4090	97
11-Sep-95	40-40%	0	0.01	141	129	128	175	22994	0.2008	40	0.3006	58
10-Sep-95	40-40%	0	0.10	67	80	81	28	7307	0.2587	56	0.3366	74
10-Sep-95	40-40%	0	0.20	59	71	72	25	6031	0.2649	59	0.3382	76
13-Sep-95	50-50%	0	0.01	141	129	128	175	77148	0.2021	43	0.2976	63
13-Sep-95	50-50%	0	0.10	67	80	81	28	24073	0.2548	60	0.3283	78
12-Sep-95	50-50%	0	0.20	59	71	72	25	34907	0.2601	61	0.3310	78

4 Networks
K-Means (with P = 10 and Normalized Gaussian) (expected number of centers = 200)

Date	Scaling	λ	Orth. Accuracy	Final centers				Training Time	Training RMS	Training % Error	Testing RMS	Testing % Error
21-Aug-95	30-30%	0	0.01	141	129	128	175	34106	0.1970	39	0.2892	53
22-Aug-95	30-30%	0	0.10	67	80	81	28	7355	0.2541	59	0.3306	75
22-Aug-95	30-30%	0	0.20	59	71	72	25	9014	0.2584	62	0.3412	76
22-Aug-95	30-30%	0	0.30	52	62	63	22	30240	0.2655	65	0.3365	82
22-Aug-95	30-30%	0	0.40	45	54	54	19	26819	0.2710	64	0.3332	84
23-Aug-95	30-30%	0	0.50	37	45	45	16	6755	0.2768	68	0.3286	84
23-Aug-95	30-30%	1	0.01	141	129	128	175	25893	0.3265	100	0.3316	89
23-Aug-95	30-30%	1	0.10	67	80	81	28	6536	0.3529	98	0.3613	93
23-Aug-95	30-30%	1	0.20	59	71	72	25	13521	0.3491	96	0.3607	94
23-Aug-95	30-30%	2	0.01	141	129	128	175	31265	0.4133	100	0.3933	100
23-Aug-95	30-30%	2	0.10	67	80	81	28	7179	0.4095	99	0.4011	97
23-Aug-95	30-30%	2	0.20	59	71	72	25	6062	0.3923	98	0.3948	97
10-Sep-95	40-40%	0	0.01	141	129	128	175	29069	0.2011	40	0.2946	55
11-Sep-95	40-40%	0	0.10	67	80	81	28	18316	0.2524	58	0.3352	80
11-Sep-95	40-40%	0	0.20	59	71	72	25	20127	0.2584	60	0.3347	80
12-Sep-95	50-50%	0	0.01	141	129	128	175	28547	0.1954	38	0.3052	62
12-Sep-95	50-50%	0	0.10	67	80	81	28	19800	0.2474	60	0.3291	75
12-Sep-95	50-50%	0	0.20	59	71	72	25	26655	0.2512	60	0.3261	75

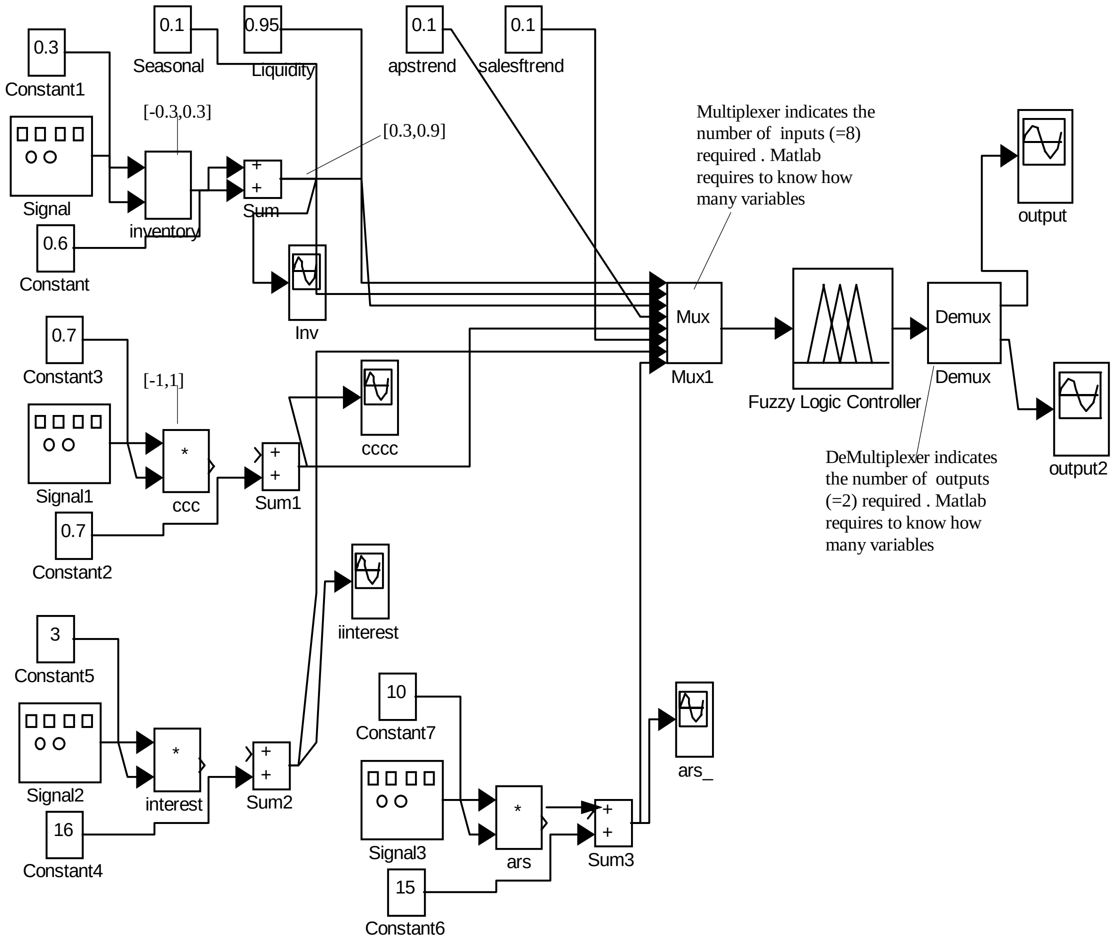
4 Networks
First-M Candidates (expected number of centers = 200)

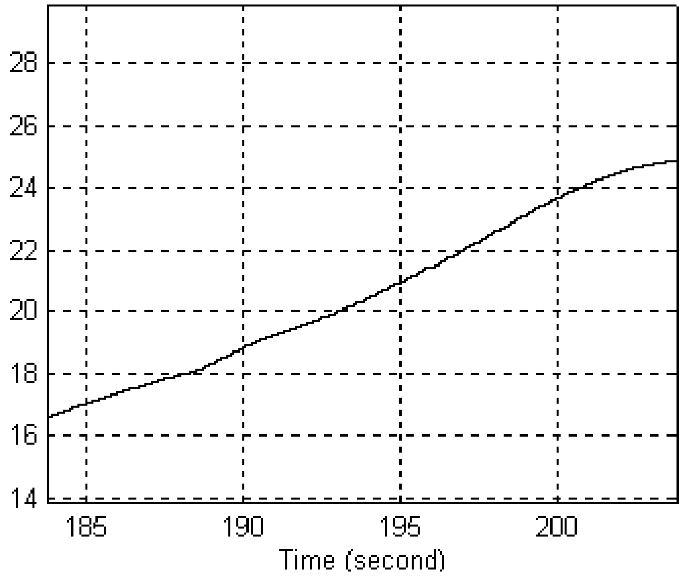
Date	Scaling	λ	Orth. Accuracy	Final centers				Training Time	Training RMS	Training % Error	Testing RMS	Testing % Error
15-Sep-95	30-30%	0	0.10	21	32	53	88	15293	0.2123	40	0.2501	50
15-Sep-95	30-30%	0	0.20	2	2	7	45	9587	0.2600	57	0.2418	58
15-Sep-95	40-40%	0	0.10	17	27	50	91	13828	0.2115	40	0.2508	47
15-Sep-95	40-40%	0	0.20	2	3	8	46	7993	0.2639	57	0.2499	59
18-Sep-95	50-50%	0	0.01	139	142	142	180	31775	0.1535	15	0.3194	65
15-Sep-95	50-50%	0	0.10	14	33	50	80	30235	0.2116	40	0.2512	53
15-Sep-95	50-50%	0	0.20	3	4	7	40	10581	0.2718	66	0.2665	73
18-Sep-95	60-60%	0	0.01	132	147	143	171	23721	0.1529	15	0.3103	62
15-Sep-95	60-60%	0	0.10	15	32	42	80	16241	0.2089	36	0.2406	45
15-Sep-95	60-60%	0	0.20	3	6	8	46	6044	0.2647	66	0.2499	68

7.6.

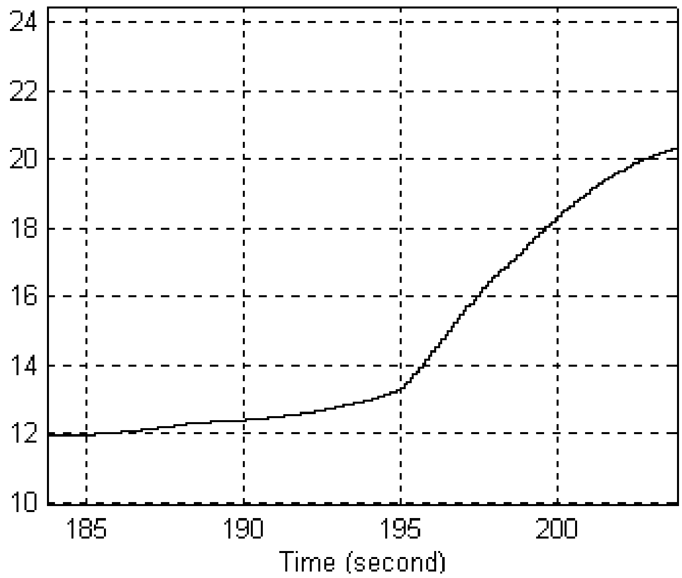
Appendix F:

Example of the Use of Simulink During the Test of Different Inputs to the Fuzzy Expert System





Reduce of Inventory



Reduce ARS (Accounts Receivable)

7.7. Appendix G:
Examples of the Rules Used in the Fuzzy Expert System (in [MART96])

1)	IF	LIQUIDITY
	and	*Seasonal* is no
	and	**Inventory** is HIGH
	and	*<Purchases> in days of Sales trend* is increasing or stable
	and	**CCC** is HIGH
	and	*Sales forecast trend* is stable or decreasing
	THEN	**Reduce Inventories** is HIGH
2)	IF	LIQUIDITY
	and	*Seasonal* is no
	and	**Inventory** is HIGH
	and	*<Purchases> in days of Sales trend* is unstable
	and	**CCC** is HIGH
	and	*Sales forecasts trend* is stable or decreasing
	THEN	**Reduce Inventories** is MEDIUM
3)	IF	LIQUIDITY
	and	*Seasonal* is no
	and	**Inventory** is MEDIUM
	and	*<Purchases> in days of Sales trend* is *increasing* or stable
	and	**CCC** is HIGH
	and	*Sales forecast trend* is stable or decreasing
	THEN	**Reduce Inventories** is MEDIUM
4)	IF	LIQUIDITY
	and	*Seasonal* is no
	and	**Inventory** is LOW
	and	*<Purchases> in days of Sales trend* is stable
	and	**CCC** is HIGH
	and	*Sales forecast trend* is stable or decreasing
	THEN	**Reduce Inventories** is LOW
5)	IF	LIQUIDITY
	and	*Seasonal* is no
	and	**Inventory** is LOW
	and	*<Purchases> in days of Sales trend* is stable
	and	**CCC** is HIGH
	and	*Sales forecast trend* is stable or decreasing
	THEN	**Reduce Inventories** is LOW
6)	IF	LIQUIDITY
	and	*Seasonal* is no
	and	**Interest** is HIGH
	and	**Accounts Receivable in Days of Sales** is HIGH
	and	**Inventory** is HIGH
	THEN	**Reduce Inventories** is HIGH
	and	**Reduce Accounts Receivable** is HIGH

7) IF LIQUIDITY
and *Seasonal* is no
and **Interest** is HIGH
and **Accounts Receivable in Days of Sales** is MEDIUM
and **Inventory** is MEDIUM
and *Sales forecast trend* is stable or decreasing
THEN **Reduce Inventories** is MEDIUM
and **Reduce Accounts Receivable** is MEDIUM

8) IF LIQUIDITY
and *Seasonal* is no
and **Interest** is MEDIUM
and **Accounts Receivable in Days of Sales** is HIGH
and **Inventory** is HIGH
and *Sales forecast trend* is stable or decreasing
THEN **Reduce Inventories** is HIGH
and **Reduce Accounts Receivable** is MEDIUM

9) IF LIQUIDITY
and *Seasonal* is no
and **Interest** is MEDIUM
and **Accounts Receivable in Days of Sales** is HIGH
and **Inventory** is HIGH
and *Sales forecast trend* is unstable
THEN **Reduce Inventories** is MEDIUM
and **Reduce Accounts Receivable** is MEDIUM

10) IF LIQUIDITY
and *Seasonal* is no
and **Interest** is MEDIUM
and **Accounts Receivable in Days of Sales** is MEDIUM
and **Inventory** is MEDIUM
and *Sales forecast trend* is decreasing
THEN **Reduce Inventories** is MEDIUM
and **Reduce Accounts Receivable** is LOW

11) IF LIQUIDITY
and *Seasonal* is no
and **Interest** is LOW
and **Accounts Receivable in Days of Sales** is HIGH
and **Inventory** is HIGH or MEDIUM
and *Sales forecast trend* is stable or decreasing
THEN **Reduce Inventories** is MEDIUM
and **Reduce Accounts Receivable** is MEDIUM

12) IF LIQUIDITY
and *Seasonal* is no
and **Interest** is LOW
and **Accounts Receivable in Days of Sales** is MEDIUM
and **Inventory** is HIGH
and *Sales forecast trend* is unstable
THEN **Reduce Inventories** is LOW
and **Reduce Accounts Receivable** is LOW

13) IF LIQUIDITY
and *Seasonal* is yes

and **Inventory** is VERY HIGH
and *Sales forecast trend* is decreasing
THEN **Reduce Inventories** is MEDIUM

14) IF LIQUIDITY
and *Seasonal* is yes
and **Inventory** is VERY HIGH
and *Sales forecast trend* is unstable
THEN **Reduce Inventories** is LOW

15) IF LIQUIDITY
and *Seasonal* is yes
and **Inventory** is HIGH
and *Sales forecast trend* is decreasing
THEN **Reduce Inventories** is LOW

16) IF LIQUIDITY
and *Seasonal* is yes
and **Interest** is HIGH
and **Accounts Receivable in Days of Sales** is VERY HIGH
and **Inventory** is VERY HIGH
THEN **Reduce Inventories** is MEDIUM
and **Reduce Accounts Receivable** is MEDIUM

17) IF LIQUIDITY
and *Seasonal* is yes
and **Interest** is HIGH
and **Accounts Receivable in Days of Sales** is VERY HIGH
and **Inventory** is HIGH
THEN **Reduce Inventories** is LOW
and **Reduce Accounts Receivable** is LOW

18) IF LIQUIDITY
and *Seasonal* is yes
and **Interest** is MEDIUM
and **Accounts Receivable in Days of Sales** is VERY HIGH
and **Inventory** is HIGH or VERY HIGH
THEN **Reduce Inventories** is LOW
and **Reduce Accounts Receivable** is LOW

19) IF LIQUIDITY
and *Seasonal* is yes
and **Interest** is MEDIUM
and **Accounts Receivable in Days of Sales** is HIGH
and **Inventory** is HIGH
and *Sales forecast trend* is decreasing
THEN **Reduce Inventories** is MEDIUM
and **Reduce Accounts Receivable** is LOW

20) IF LIQUIDITY
and *Seasonal* is yes
and **Interest** is MEDIUM
and **Accounts Receivable in Days of Sales** is HIGH
and **Inventory** is HIGH
and *Sales forecast trend* is unstable
THEN **Reduce Inventories** is LOW
and **Reduce Accounts Receivable** is LOW

21) IF LIQUIDITY
and *Seasonal* is yes
and **Interest** is LOW
and **Accounts Receivable in Days of Sales** is HIGH
and **Inventory** is HIGH or MEDIUM
and *Sales forecast trend* is stable or decreasing
THEN **Reduce Inventories** is LOW
and **Reduce Accounts Receivable** is LOW

22) IF LIQUIDITY
and **Accounts Payable in Days of Sales** is LOW
THEN *Negotiate Payable terms*

23) IF LIQUIDITY
and *Seasonal* is no
and **STDebt** is HIGH
and **Interest** is HIGH
and T / | WI | trend is negative
and **Sales Volume** is HIGH
and *business phase* is growth
THEN *Slow down business growth*

24) IF LIQUIDITY
and *Seasonal* is no
and **STDebt** is HIGH
and **Levarage** is HIGH
and **Operational Income** is HIGH
THEN *Inject Equity*

25) IF LIQUIDITY
and DEBT is no
and *Seasonal* is no
and *mid-term Sales forecast trend* is stable or increasing
and **STDebt** is HIGH or MEDIUM
and **Levarage** is LOW
and **Operational Income** is HIGH
THEN *Increase LTDebt position*

26) IF PROFITABILITY
and **Sales Volume** is LOW
and **Contribution Margin** is HIGH
and **Selling Expenses** is HIGH
THEN **Reduce Selling Expenses** is HIGH

27) IF PROFITABILITY
and **Sales Volume** is LOW
and **Contribution Margin** is HIGH
and **Average Cost of Purchases** is HIGH
THEN **Reduce Cost of Purchases** is HIGH

28)IF PROFITABILITY
and **Sales Volume** is LOW
and **Contribution Margin** is HIGH
and **Average Cost of Purchases** is AVERAGE
and **Selling Expenses** is HIGH
THEN **Reduce Cost of Purchases** is LOW

29) IF PROFITABILITY
and **Sales Volume** is LOW
and **Contribution Margin** is HIGH
and **Average Cost of Purchases** is AVERAGE
and **Selling Expenses** is AVERAGE
THEN **Reduce Cost of Purchases** is MEDIUM

30) IF PROFITABILITY
and **Sales Volume** is LOW
and **Contribution Margin** is HIGH
and *elasticity* is yes
and **Average Cost of Purchases** is LOW
and **Selling Expenses** is LOW
THEN **Reduce Margin** is MEDIUM

31) IF PROFITABILITY
and **Sales Volume** is LOW
and **Contribution Margin** is HIGH
and *elasticity* is no
and **Average Cost of Purchases** is LOW
and **Selling Expenses** is LOW
THEN **Reduce Margin** is LOW

32) IF PROFITABILITY
and **Sales Volume** is LOW
and **Contribution Margin** is HIGH
and *elasticity* is yes!
and **Average Cost of Purchases** is LOW
and **Selling Expenses** is LOW
THEN **Reduce Margin** is HIGH

33) IF PROFITABILITY
and **Sales Volume** is HIGH
and **Contribution Margin** is LOW
and *elasticity* is yes!
THEN **Increase Margin** is MEDIUM

34) IF PROFITABILITY
and **Sales Volume** is MEDIUM
and **Contribution Margin** is LOW
and *elasticity* is yes!
THEN **Increase Margin** is LOW

35) IF PROFITABILITY
and **Administrative Expenses** is HIGH
THEN **Reduce Administrative Expenses** is HIGH

36) IF PROFITABILITY
and **Administrative Expenses** is MEDIUM
THEN **Reduce Administrative Expenses** is LOW

37) IF PROFITABILITY
and **Contribution Margin** is HIGH
THEN *Check pricing in relation to product turnover*

38) IF PROFITABILITY
and *Seasonal* is yes
and **Interest** is HIGH

and **Inventory** is VERY HIGH OR HIGH
THEN **Reduce Inventories** is MEDIUM

39) IF PROFITABILITY
and *Seasonal* is yes
and **Interest** is HIGH
and **Inventory** is MEDIUM
THEN **Reduce Inventories** is LOW

40) IF PROFITABILITY
and *Seasonal* is no
and **Interest** is HIGH
and **Inventory** is VERY HIGH OR HIGH
THEN **Reduce Inventories** is MEDIUM

41) IF PROFITABILITY
and *Seasonal* is no
and **Interest** is (HIGH or MEDIUM)
and **Inventory** is VERY HIGH OR HIGH
THEN **Reduce Inventories** is HIGH

42) IF PROFITABILITY
and *Seasonal* is no
and **Interest** is (VERY HIGH OR HIGH or MEDIUM)
and **Inventory** is MEDIUM
THEN **Reduce Inventories** is MEDIUM

43) IF PROFITABILITY
and **Interest** is LOW
and **Inventory** is HIGH or VERY HIGH or MEDIUM
THEN **Reduce Inventories** is MEDIUM

44) IF PROFITABILITY
and **Interest** is HIGH
and **Accounts Receivable in Days of Sales** is VERY HIGH
THEN **Reduce Accounts Receivable** is HIGH

45) IF PROFITABILITY
and *Seasonal* is yes
and **Interest** is HIGH
and **Accounts Receivable in Days of Sales** is HIGH
THEN **Reduce Accounts Receivable** is MEDIUM

46) IF PROFITABILITY
and **Interest** is MEDIUM
and **Accounts Receivable in Days of Sales** is VERY HIGH
THEN **Reduce Accounts Receivable** is MEDIUM

47) IF PROFITABILITY
and *Seasonal* is no
and **Interest** is MEDIUM
and **Accounts Receivable in Days of Sales** is HIGH
THEN **Reduce Accounts Receivable** is MEDIUM

48) IF PROFITABILITY
and LIQUIDITY is no
and *mid-term Sales forecast trend* is stable or increasing
and **Leverage** is LOW

and *Interest mid-term forecast trend* is stable or decreasing
and **Operational Income** is HIGH
THEN *Increase Leverage by LTDebt*

49) IF DEBT
and *mid-term Sales forecast trend* is stable or decreasing
and **Fixed Assets Turnover** is LOW
THEN *Reduce Investments in Fixed Assets*

50) IF DEBT
and PROFITABILITY is no
and *mid-term Sales forecast trend* is stable or decreasing
and **Interest** is (HIGH and *Interest mid-term forecast trend* is stable or decreasing)
and Leverage is HIGH
THEN *Negotiate debt terms* and *inject Equity*

51) IF DEBT
and *mid-term Sales forecast trend* is stable or decreasing
and **Interest** is HIGH and *Interest mid-term forecast trend* is increasing
and **Leverage** is HIGH
THEN (*Negotiate debt terms* or *inject Equity*) and *Stop borrowing*

52) IF DEBT
and *mid-term Sales forecast trend* is stable or decreasing
and **Interest** is HIGH and *Interest mid-term forecast trend* is stable or decreasing
and Leverage is MEDIUM
THEN *Negotiate debt terms*

53) IF DEBT
and **Sales Volume** is (LOW or MEDIUM)
and **ST I Expense / LT I Expense** is MEDIUM or HIGH
and **Interest** is HIGH or MEDIUM and *Interest mid-term forecast trend* is stable or decreasing
THEN *Restructure terms between LT and ST Debt*

54) IF DEBT
and PROFITABILITY is no
and **Sales Volume** is HIGH
and **ST I Expense / LT I Expense** is MEDIUM or HIGH
and *Interest mid-term forecast trend* is increasing
THEN *Increase Debt payment*

55) IF DEBT
and **Operational Cash Flow** is HIGH
THEN **Reduce Fixed Costs** is MEDIUM

56) IF DEBT
and **Operational Cash Flow** is MEDIUM
THEN **Reduce Fixed Costs** is MEDIUM

57) IF DEBT
and **Levarage** is HIGH or MEDIUM
and **Interest** is HIGH or VERY HIGH
and *Credit Line* is restricted
THEN **Reduce Fixed Costs** is HIGH

58) IF DEBT
and **Levarage** is MEDIUM
and **Interest** is MEDIUM

and *Credit Line* is available
THEN **Reduce Fixed Costs** is MEDIUM

59) IF DEBT
and **Levarage** is HIGH or MEDIUM
and **Interest** is HIGH or VERY HIGH

THEN **Reduce Inventories** is HIGH
and **Reduce Accounts Receivable** is MEDIUM

60) IF DEBT
and **Levarage** is HIGH
and **Interest** is LOW
THEN **Reduce Inventories** is MEDIUM
and **Reduce Accounts Receivable** is LOW

61) IF DEBT
and **Interest** is HIGH
and **Fixed Charge Coverage** is LOW
and *Sales forecast trend* is decreasing
THEN *Negotiate Lease terms*

62) IF DEBT
and PROFITABILITY is no
and **ST I Expense / LT I Expense** is MEDIUM or HIGH
and *mid-term Sales forecast trend* is stable or decreasing
THEN *Restructure terms between LT and ST Debt (Extend Debt terms)*

63) IF DEBT
and LIQUIDITY
and PROFITABILITY is no
and **Interest** is HIGH or MEDIUM
and **Contribution Margin** is AVERAGE or HIGH
and **Sales Volume** is HIGH
THEN *Increase On Sale offers*

64) IF DEBT
and LIQUIDITY
and PROFITABILITY is no
and **Interest** is HIGH or MEDIUM
and **Contribution Margin** is LOW
THEN *Redefine On Sale mix*

7.8. Appendix H:
MATLAB Code of the Backpropagation Network

```
% PROGRAM: backprop()
% Backpropagation Network to Diagnose Financial Health Problems
% TECHNICAL OBSERVATION:

% This program implements 3 Backpropagation Algorithms
%    Roberto Pacheco and Alejandro Martins, July 1995.
clc
disp('============================================================')
disp(' BACKPROPAGATION TO CLASSIFY THE COMPANIES - TRAIN AND TEST ')
disp('============================================================')
echo off;
t=clock;
%%%%%%%%%%%%%%%%%% Reading Network Parameters %%%%%%%%%%%%%%%%%%
typeBP =-1;
while (typeBP < 1) | (typeBP > 3),
   disp('Type of Backpropagation Optimizing Algorithm: ');
              disp('    [1] Simple BP - poor');
              disp('    [2] Optimized BP            - reasonable approximation ');
              disp(' [3] Levenberg-Marquardt BP          - excellent but very slow');
              typeBP = input('Chose One: ');
end;
totHidden = input ('Total of Hidden Neurons: ');
max_epoch = input('Maximum Number of Epochs: ');
RMSgoal = input('RMS Goal: ');
typeAFH = -1; typeAFO=-1;
while ((typeAFH < 49) | (typeAFH > 51))&((typeAFO < 49) | (typeAFO > 51)),
disp('Types of Activation Function ');
disp(' [1] logaritmic  sigmoid-logsig (interval[0,1])');
disp(' [2] tangent sigmoid     - tansig (interval [-1,1])');
disp(' [3] linear                    - purelin (interval [-1,1])');
str = input('Chose: (for Hidden) <space> (for Output): ','s');
typeAFH = str(1); typeAFO = str(3);
end;
if(typeAFH == 49)
actFunHL = 'logsig'; %%% activation function in the hidden layers
elseif (typeAFH == 50)
actFunHL = 'tansig'; %%% activation function in the hidden layers
else
actFunHL = 'purelin'; %%% activation function in the hidden layers
end;
if(typeAFO == 49)
actFunOL = 'logsig'; %%% activation function in the output layers
elseif (typeAFO == 50)
actFunOL = 'tansig'; %%% activation function in the output layers
else
actFunOL = 'purelin'; %%% activation function in the output layers
end;
clear str typeAFH typeAFO;
clc; clc;
disp( ==============================================================')
disp(' BACKPROPAGATION TO CLASSIFY THE COMPANIES - TRAIN AND TEST' )
disp( ==============================================================')
disp(' ');

fprinff(sprinff('Number of Layers              : 2')); disp('');
fprinff(sprinff('Number of Hidden Neurons : %%d'),totHidden); disp('');
fprinff(sprinff('Hidden Activation Function  : %%s'),actFunHL); disp('');
fprinff(sprinff('Output Activation Function   : %%s'),actFunOL); disp('');
fprinff(sprinff('RMS Goal                          : %%1.3f'),RMSgoal); disp('');
msg = sprinff('Backpropagation Algorithm  : %%s');
```

```
if (typeBP==1 )
   fprinff(msg,'Simple BackProp (Grad. Desc. without momentum and Adaptive Lr)');
elseif(typeBP==2)
   fprinff(msg,'Optimized Backpropagation (Grad. Desc. with momentum + adap. Lr)');
elseif(typeBP==3)
  fprinff(msg,'Levenberg-Marquardt Optimization');
end;
ch = -1;
while (ch ~= 1) & (ch ~= 2)
  disp(' '); disp('');
  ch = input('Press any [1] to initiate training or [2] to cancel: ');
end;
if (ch==2)
  return
end;
clear ch;
clc;
echo off;
disp(' ');disp('');
fprinff(' TRAINING...');

%%%%%%%%%%%%%%%%% Loading Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('');
fprinff('                  1. Loading Training Files...');
load train.dat;   load scales.dat        %%%% scale factors to the scaling of the input vectors
y          = train(:,10:13); %%%% outputs to the training set.
%%%%%%%%%%%%%%% Testing the dimensions %%%%%%%%%%%%%%%%%%%%%%%
[N p]       = size(train);
[lineY Q] = size(y);
if (lineY ~= N)
  disp('Error: the total of targets is inferior to the total of inputs');
  retum;
end;
echo off; clear p lineY;


%%%%%%%%%%%% FIRST STEP: SCALING OF THE INPUT VECTORS %%%%%%%%%%%%%%%%
disp('');    fprinff('                      2.Scaling Inputs...');
load mins.dat: load maxs.dat:
x = scale(train(:,2:9),min(train(:,2:9)),max(train(:,2:9)),mins,maxs, scales;
[N P] = size(x);

%%%%%%%%%% transforming 'y' into .1 or .9 vectors %%%%%%%%%%%%%%%%%%%%%%
disp(''):
fprinff('      3. Transforming outputs              ');
for i=1 :N
              for j=1 :Q
  if (y(i,j) == 0)
                  y(i,j)=0.1;
              else
                   Y(i,j)=0.9;
              end;
 end;
end;

%%%%%%% SECOND STEP: ADJUSTING LEARNING AND TRAINING PARAMETERS %%%%%%
Tolerance = 0.3;
disp_freq=1;
lr=0.95;
momentum=0.95;
err_ ratio=1.04;
tp = [disp_freq max_epoch RMSqoal lr momentum err_ratio];

%%%%%%% THIRD STEP: INITIALIZING THE NETWORK %%%%%%%%%%%%%%%%%%%%%%
Pmin max=[min(x);max(x)]';
% Creates a network with "totHidden" neurons with "actFunHL"
```

```
% in the first layer and "y" dimension neurons with
% a "actFunOL" activation function in the second layer.
[W1,b1,W2,b2] = initff(PminMax,totHidden,actFunHL,4,actFunOL);

%%%%%%% FOURHT STEP: TRAINING THE NETWORK %%%%%%%%%%%%%%%%%%%
disp('');fprinH('                    4. Training outputs    '); disp(''); disp('');
if (typeBP ==1) %%% Classical Backpropagation
[W1,b1,W2,b2,te,tr]=trainbp(W1 ,b1 ,actFunHL,W2,b2,actFunOL...
,x',y',tp);
elseif (typeBP == 2) %%% Optimized Backpropagation
[W1,b1,W2,b2,te,tr]=trainbpx(W1,b1 ,actFunHL,W2,b2,actFunOL...
 ,x',y',tp);
elseif (typeBP == 3) %%% Levenberg-Marquardt Backpropagation
[W1,b1,W2,b2,te,tr]=trainlm(W1,b1,actFunHL,W2,b2,actFunOL...
        ,x',y',tp);
end;
t = etime(clock,t);
clear disp_freq err_ratio i j tp;     pause;

disp(''); fprinff('        Results:');disp(' ');

%%%%%%% FIFTH STEP: VERIFYING TRAINING PERFORMANCE %%%%%%%%%%%%
YN = simuff(x',W1,b1,actFunHL,W2,b2,actFunOL)';
Error = y - YN;

%%%%%%%%% RMS in the Training Phase
RMSTrain = sqrt(sum(sum(Error.^2))/(N*Q));
fprinff(sprintf('                    RMSTraining            : %%2.4f'),RMSTrain); disp('');

%%%%%%% Percentual Error in the Training Phase
PErrTrain = errperc(y,YN,Tolerance);
fprinff(sprinff('                    Perceptual Error: %%2.2f %'),PErrTrain); disp('');

disp('');disp(''); disp('                              Press any Key to Continue ');pause;

%%%%%%% SIXTH STEP: TESTING PERFORMANCE %%%%%%%%%%%%%%%%%%%
clc; disp(' ');
fprinff('  TESTING...'); disp(' ');
fprinff('        1. Loading Testing Files...');disp(' ');
load testing.dat;
y = testinq(:,10:13); %%%% outputs to the training set.

%%%%%%%%%%%% testing dimensions
[N QT] = size(y);
jf (Q ~= QT)
disp('Error: the testing data is not related to this network - output dimensions do not match!');
 return;
end;

%%%%%%%%%%%%% scaling the input vector
fprinff('     2. Scaling  ');disp(' ');
x = scale(testing(:,2:9),min(train(:,2:9)),max(train(:,2:9)),mins, maxs,scales);
[N PT] = size(x);
clear scales train;
if (PT ~= P)
  disp ('Error: testing and training input patterns have different dimensions!');
  return;
end;
clear QT PT;

%%%%%%%%%% transforming 'y' into .1 or .9 vectors
fprinff('     3. Transforming outputs           ');disp(' ');
for i=1 :N
for j=1 :Q
        if (y(i,j) == 0)
   y(i,j)=0.1;
```

```
            else
                y(i,j) = 0.9;
     end;
  end;
end;

%%%%%%%%% Network Response to the Testing Inputs
YN = simuff(x',W1,b1,actFunHL,W2,b2,actFunOL)';
Error = y - YN;

disp(''); disp('');fprinff('                    Results:');disp(' ');
%%%%%%%%% RMS in the Testing Phase
RMSTest = sqrt(sum(sum(Error.^2))/(N*Q));
fprinff(sprinff('            RMS Testing                          : %%2.4f ',)RMST); disp(' ');

%%%%%%% Percentual Error in the Testing Phase
PErrTest = errperc(y,YN,Tolerance);
fprinff(sprintf('                Percentual Error                  : %%2.2f%'),PerrTest);
disp('');disp(''); disp(''); disp('');disp(''); disp('');
disp('Press any Key to Continue ');pause;

clc; clc;
disp('=================================================================')
disp(' BACKPROPAGATION TO CLASSIFY THE COMPANIES - TRAIN AND TEST')
disp('=================================================================')
fprinff(sprinff('Date                                         : %%s'),date);disp('');
fprinff(sprinff('Number of Layers                  : 2')); disp('');
fprinff(sprinff('Number of Hidden Neurons       : %%d'),totHidden); disp('');
fprinff(sprinff('Hidden Activation Function        :%°/Os'),actFunHL); disp('');
fprinff(sprinff('Output Activation Function         : %%s'),actFunOL); disp('');
fprinff(sprinff('RMS Goal              : %%1.3f'),RMSgoal); disp('');
msg = sprinff('Backpropagation Algorithm        : %%s');
if (typeBP==1)
fprinff(msg,'Simple BackProp (Grad. Desc. without momentum and Adaptive Lr)');
elseif(typeBP==2)
fprinff(msg,'Optimized Backpropagation (Grad. Desc. with momentum + adap. Lr)');
elseif(typeBP==3)
fprinff(msg,'Levenberg-Marquardt Optimization');
end;
disp(' ');disp('            TRAINING Results:');
fprintf(sprinff('                    No Epochs : %%3d'),te); disp('');
fprinff(sprinff('                    Training Time : %%2.4f seconds'),t); disp('');
fprinff(sprinff('                    RMS Training : %%2.4f'),RMSTrain); disp('');
fprinff(sprinff('                    Percentual Error: %%2.2f'),PErrTrain);
fprinff('%%');
disp(' ');disp('            TESTING Results: ');
fprinff(sprinff('            RMS Testing : %%2.4f'),RMSTest); disp('');
```

7.9. Appendix I:
MATLAB Code of the RBF Network

```
% PROGRAM: ortrbf()
% Orthogonal Least Squares Learning Algorithm for RBF
%    Chen, Cowan, and Grant
%    IEEE Trans. on NN, Vol. 2, No. 2 -1991.
%
% TECHNICAL OBSERVATION:
%........................................
% This program implements a Radial Basis Function Network with
% the orthogonal least squares learning algorithm.
% Roberto Pacheco and Alejandro Martins, July 1995.
% Variables:
% x is (N x p)             : input vectors.
% y is (N x q)             : target vectors.
% M          : expected number of centers
% CWNorms is q x p  : weights for each weighted matrix
% RMSgoal              : precision (e.g., 0.003)
% Ms(i)     : actual number of centers for output'i'
% cent is (N x sum(Msi))            : vectors of centers
% radius is(N x sum(Msi))           : the radius of the centers
clc; clc;
disp('=================================================================')
disp(' SUPERVISED (Ort) RBF TO CLASSIFY THE COMPANIES - TRAIN AND TEST' )
disp('=================================================================')
echo off;   t=clock;
%%% Tolerance to the difference between target and network responses
Tolerance = 0.30;
load train.dat:
load wnomms.dat;   %%%% Weights to the weighted norm
load scales.dat       %%%% scale factors to the scaling of the input vectors
y            = train(:,10:13); %%%% outputs to the training set.
NORMS   = wnorms;             clear wnorms;
M           = input('Expected Number of Centers: ');
%%% PO - error maximum associated to the regression (choice of the centers)
PO = input('Approximation Grade in Orthonormalization Process: ');
typeRBF =-1;
while (typeRBF < 0) | (typeRBF > 5),
   disp('Type of Radial Basis Function: ');
   disp('      [1] Gaussian     [2] Multiquadratic    [3] Inverse Multiquadratic');
   disp('      [4] Cubic Spline [5] Linear Spline');
   typeRBF = input('Chose One: ');
   if (typeRBF > 5)
                 typeRBF = 0;
  end;
end;
if (typeRBF==1)        tRBF =Gaussian';
  elseif(typeRBF==2)  tRBF= 'Multiquadratic';
  elseif(typeRBF==3)  tRBF = 'Inverse multiquadratic';
  elseif(typeRBF==4)  tRBF = 'Cubic Spline';
  else                        tRBF = 'Linear Spline';
end;
typeNetwork = -1;
while (typeNetwork ~= 0) & (typeNetwork ~= 1),
     disp('Type of RBF Network: [0] Non-normalized       [1] Normalized');
     typeNetwork = input('Chose One: ');
end;
it (typeNetwork==0) tNet = 'Non-normalized';
             else                                   tNet = 'Normalized';
end;
typeDetCent = -1;
while (typeDetCent ~= 0) & (typeDetCent ~= 1),
```

```
    disp('Method of Center Determination: [0] First-M Centers       [1] K-means');
    typeDetCent = input('Chose One: ');
end;
if(typeDetCent)
        tDetCent = 'K-means';
        PN = input('P in the P-nearest neighbor Algorithm: ');
else
   tDetCent = 'First-M Centers';
end;
Lambda = input('Regularization Parameter: ');
RMSgoal = input('RMS goal : ');
clc; clc
disp('=================================================================')
disp(' SUPERVISED (Ort) RBF TO CLASSIFY THE COMPANIES - TRAIN AND TEST' )
disp('=================================================================')
fprinff(sprinff('Expected Number of Centers: %%d'),M); disp('');
fprinff(sprinff('Accuracy in Orthonormalization            : %%2.3f'),PO);disp('');
fprinff(sprinff('RMS Goal            : %%1.3f'),RMSgoal); disp('');
msg = sprinff('Type of Radial Basis Function              : %%s');
fprinff(msg,tRBF); disp(' ');
msg = sprintf('Type of Radial Basis Network:%%s');
fprinff(msg,tNet); disp('');
msg = sprinff('Method of Center Determination             : %%s');
fprinff(msg,tDetCent);
if(typeDetCent)
   msg=sprinff(' (P = %%2.2f)');fprinff(msg,PN);
end;
disp(''); fprinff(sprinff('Regularization Parameter : %%3.2f (Lambda)'),Lambda); echo off;
disp(' ');disp(''); fprinff(' TRAINING...');
%%%%%%%%%%%%%% Testing the dimensions %%%%%%%%%%%%%%%%%
[N p]       = size(train);
[lineY Q]  = size(y);
[lineCW colCW] = size(NORMS);
if (lineY ~= N)
   disp('Error: the total of targets is inferior to the total of inputs');
  return;
end;
if (lineCW ~= Q)
  disp ('Error: there must be the same number of weighted matrices as there are outputs!');
  retum;
end;
echo off;
clear p lineY lineCW;
%%%%%%% transforming 'y' into -1 or 1 vectors
disp(' '); fprinff(' 1. Transforming outputs       ');
for i=1 :N
  for j=1 :Q
          if (y(i,j) <= 0)
            y(i,j)=0.1;
         else
            Y(i.i) = 0 9;
           end;
 end;
end:

%%%%%%%%%%% FIRST STEP: SCALING OF THE INPUT VECTORS %%%%%%%%%%%%%
disp(' ');
fprinff('2. Scaling...');disp('');
x = train(:,2:9);
maxT = max(train(:,2:9)); minT = min(train(:,2:9));
x = scale(x,minT,maxT,scales);
varT = var(x);
[N P] = size(x);
clear train;
if (colCW~= P)
  disp ('Error: there must be the same number of weights in the Weighted matrices as there are inputs!');    retum;
```

```
end;
clear colCW;

%%%%%% SECOND STEP: FORMATION OF THE (Q)-SETS OF CENTERS %%%%%%%%%%
   disp(' ');        fprinff('                  3. Forming the Center Candidates...');
CENTS = []; %%% matrix with the Q-sets of M candidates to be centers
RADS = []; %%% matrix with the Q sets of radius of the M centers
begCent = 1;
endCent = M;
for phase = 1:Q
 %////////////// CWNorm of the input patterns with respect to the Output coordinate "phase"
 CWNorm = sqrt(P*diag(NORMS(phase,:))); %% each diagonal element is equal to sqrt(P*wi)
 %%%%% Initialization of the M expected Centers
 if (typeDetCent ==0)
   %//////////// The first M input patterns are candidates to be Centers
   CENTS(:,begCent:endCent) = x(1 :M,:)'; % initial M centers
   %////////////// Radius of the centers
   % radius = averdist(x,CENTS(:,begCent:endCent)',CWNorm); % M center radius
 r = 2*P*sum(NORMS(phase,:)./varT);
 radius = ones(1,M)*r;
 %%%%%% Add the radius of the "phase"-th set of Centers
 RADS = [RADS';radius']';
 else
    %/////////// The candidates to be Centers are determined by the Kmeans Algorithm
    [cents rads] = kmeanwn(x,M,CWNorm); % Kmeans
    %/////////// Radius of the centers - p-nearest neighbor  Algorithm
    rads= spread(cents,PN,CWNorm)';
    CENTS = [CENTS,cents'];
    RADS = [RADS';rads']';
    clear cents r rads;
 end;
 %%%%%%% update the center counters
 begCent = endCent+1;
 endCent = endCent+M;
end;
Ms        = zeros(1,Q); % total of centers in each subgroup of the hidden layer
POSCENTS = [];           % matrix with the Q sets of Ms(1 ,phase) Positions in each set of centers
disp(' '); fprinff('                  4. Finding Orthogonal Basis - Center Selection...');
%%% FOR related to the Q Sets of Centers
begCent = 1;
endCent = M;
for phase=1:Q
  MaxErrors = [];    %%% orthogonalization error of each basis vector
  CentersPos = D; %%% positions of the formed centers (orthogonal basis) (Ms(phase) x 1)
 %////////////////CWNorm of the input patterns with respect to the Output coordinate 'phase"
 WNorm = sqrt(Ptdiag(NORMS(phase,:))); %% each diagonal element is equal to sqrt(P*wi)
 %///////////// Gaussian Matrix (Haykin pg. 259)
  maxEr = 0; error = [];
  for j=begCent:endCent
    for i=1:N
       dist = (x(i,:)-CENTS(:,j)')./sqrt(var(x));
       dist = sqrt(dist*CWNom'*CWNorm*dist');
      Gauss(i,j) = rbfunct(dist,RADSt),typeRBF);
   end;
   %% orthogonalization error of the center just obtained
   error(j) = (Gauss(:,j)'*y(:,phase))A2/(Gauss(:,j)'*Gauss(:,j));
   error(j) = error(j)/(y(:,phase)'y(:,phase));
 end;
[maxEr posBas ] = max(error);

%%%%%////////////////// Formation of the Ms Centers associated to the Nphase"th set A = Gauss(:,posBas); %% matrix with the
orthogonal basis (N x Ms)
while ((1 - sum(MaxErrors)) > PO) & (Ms(1,phase) < M)
  %/// append the current error to the MaxErrors vector
  MaxErrors = [MaxErrors;maxEr];
  %||| append the current (position) center to the formed basis
```

```
   CentersPos = [CentersPos;posBas];
   Ms(1 ,phase) = Ms(1 ,phase) + 1; %% total of centers is increased
   maxEr = 0; posBas = 0; error = [];
   for j=begCent:endCent
      if (~findvec(CentersPos',j))
         %%%%% b is a orthonormal vector
         b = gramsch(A,Gauss(:,j));
         %%%%% orthogonalization error of the center just obtained
         error(j) = (b'*y(:,phase))^2/(b'*b);
         error(j) = error(j))/(y(:,phase)'*y(:,phase));
     end;
 end; %% already tested all centers and choose the next orthonormal basis vector
if (error ~=[] )
   [maxEr posBas] = max(error);
   %llll addition of the new basis vector to the matrix of orthogonal vectors
   A = [A';gramsch(A,Gauss(:,posBas))']';
end;
end: %%% end of the formation of the Ms Centers if the "phase"th set

 %%%%% Add the center positions associated to the 'phase-th' set of Centers
 POSCENTS = [POSCENTS;CentersPos];
 %%%%%%% update the center counters
 begCent = endCent+1;
 endCent = endCent+M;
end: %%%% End of the formation of all i = 1.Q sets of (Msi) Centers
clear A b MaxErrors maxEr error CentersPos posBas radius;

%%%%%%%% THIRD STEP: FORMATION OF THE MATRIX OF CENTERS%%%%%%%%%%%
CENTERS = []; % matrix with all Q sets of Ms-centers
RADIUS = [];    % radius of the (sum(Ms(1,1 :Q)) x Q centers
begCent = 1; %%% first column of the vector with the Center positions in each 'phase' (1 to Q)
endCent = 0; %%% last column (idem)
%%% FOR related to the (Q) Sets of Centers
for phase=1:Q
  endCent = endCent + Ms(1,phase);
  PosCents = POSCENTS (begCent:endCent,1);
  for j=1:Ms(1 ,phase)
  CENTERS(begCent+j-1,:) =CENTS (:,PosCents(j)
   RADIUS(1,begCent+j-1) = RADS(1,PosCents0);
  end;
  begCent = begCent + Ms(1 ,phase);
end;
CENTERS = CENTERS';
%%%%%%%Form of the final matrix with the Q sets of center: %%
%% CENTERS becomes the following matrix:
%%
%% || 1        ..         1        |    | Q        ..   Q          ||
%% || C1,1    ..    CMs(1)1,1 |    | C1,1 .. CMs(Q)1,1   ||
%%||...         ..                   |    |  ....    ..       ...        ||
%% || 1        ..         1        |    |     Q ..          Q      ||
%% || C1,8    ..     CMs(1)1,8|    | C1,8 ..  CMs(Q)1,8  ||
%%

%%%%%% FOURTH STEP: FORMATION OF THE GAUSSIAN MATRIXES%%%%%%%%%%%%% disp(' );
fprinff('                           5. Forming Gaussian Matrixes...');
clear Gauss PosCents CENTS RADS;
GAUSSIAN = []; %%% gaussian matrix of all centers (as done in CENTERS) (N x sum(Ms(phase))
begCent = 1; %% first column of the vector with the Center positions in each “phase” (1 to Q)
endCent = 0; %%% last column (idem)
%%% FOR related to the (Q) Sets of Centers
for phase=1:Q
  endCent = endCent + Ms(1,phase);
 %/////////////// CWNorm of the input patterns with respect to the Output coordinate “phase”
 CWNorm = sqrt(P*diag(NORMS(phase,:))); %% each diagonal element is equal to sqrt(P*wi)
 %% determines the Gaussian Matrix
 for i=1:N
```

```
  for j=1:Ms(1,phase)
  rad = RADIUS(1,begCent+j-1);
  dist = (x(i,:)-CENTERS(:,begCent+j-1 )')./sqrt(varT);
  dist = sqrt(dist*CWNorm'*CWNorm*dist');
  GAUSSIAN(i,begCent+j-1 )=rbfunct(dist,rad,typeRBF);
 end;
end;
%%%%%%%%%%%%%%%%% Normalized RBF %%%%%%%%%%%%%%%%%%
if (typeNetwork)
  for i=1:N
%   tot = sum(GAUSSIAN(i,begCent:endCent));
    tot = max(GAUSSIAN(i,begCent:endCent));
      GAUSSIAN(i,begGent:endCent) = GAUSSIAN(i,begCent:endCent)/tot;
  end;
 end;
 clear tot;
 begCent = begCent + Ms(1,phase);
end; %% end of the phase.

%%%%%%%% FIFTH STEP: FORMATION OF THE WEIGHT VECTOR %%%%%%%%%%%%%
disp(' ');
fprinff('                        6. Forming the Weight Vector..');
%%% FOR related to the (Q) Sets of Centers
YN      = []; %%% network responses (is a N x Q) vector
WEIGHT = []: %%% weight matrix of the neural network (sum(Ms(:,:)) x 1 )
begCent = 1; %%% first column of the vector with the Center positions in each 'phase (1 to Q)
endCent = 0; %%% last column (idem)
for phase=1:Q
   endCent = endCent + Ms(1,phase);
   G = GAUSSIAN(:,begCent:endCent);
   CWNomm = sqrt(P*diag(NORMS(phase,:))); %% each diagonal element is equal to sqrt(P*wi)
   %% determines the Go Matrix to be used in the pseudo-inverse
   Go=[];
   for i=1: Ms(1,phase)
     for j=1: Ms(1 ,phase)
       rad = RADIUS(1,begCent+j-1);
      dist = wnormvar(CENTERS(:,begCent+i-1 )',CWNorm,CENTERS(:,begCent+j-1 )',var(x))
      Go(i,j) = rbfunct(dist,rad,typeRBF);
     end;
   end;
   %%% Inclusion of the bias Vector
   G = [G';ones(N,1)']';
   Go = [Go;ones(Ms(1,phase),1)']; %% adds a line of ones in the Go matrix
   Go = [Go';ones((Ms(1 ,phase)+1),1)'j'; %% adds a column to the Go matrix
   %%% "weight" is a column vector (Ms(phase) x 1) with the weights of the current set center
   Weight = ((inv(G'*G + Lambda*Go)*G')*y(:,phase));
   %%% The network response to the 'phase' output neuron
   YN = [YN;(G*weight)'];
   %%% WEIGHT is a column vector (sum(MS(1 ,phase) x 1 ) with the Q sets of weights
   WEIGHT = [WEIGHT;weight];
   begCent = begCent + Ms(1,phase);
end:

%%^%%%%%%%%%%% Formation of the Network response vector
YN=YN'; %%is a N x Q vector
%%/ “Purification” of the output answers (greater than (1-Tolerance) it is 0.9
%%///// and lower than Tolerance, it is = 0.1)
for i=1:N
 for j=1:Q
     if (YN(i,j) <=Tolerance)
        YN(i,j) = 0.1;
      elseif (YN(i,j) > (0.9-Tolerance))
          YN(i,j) = 0.9;
        end;
     end;
end:
```

```
%%%%%%% Formation of the last output coordinate %%%%%
%the last output coordinate is 1 when the others are "off" and 0 otherwise
for i=1:N
   YN(i,Q) = 0.9;
   for j=1:Q
        if (YN(i,j) >= 0 5)
          YN(i,Q) = 0.1;
          break;
       end;
     end;
end;

t=etime(clock,t);
disp(' ');
fprinff('        Results:');disp('');
%%%%%%%%%%%%% CALCULUS OF THE NETWORK RMS %%%%%%%%%%%%%%%
Error = y - YN;
RMSTrain = sqrt(sum(sum(Error.^2))/(N*Q));
fprinff(sprinff('            RMSTraining             : %%2.4f'),RMSTrain); disp('');
%IIIIIIIIIIIIIIIII Percentual Error of the Training
PErrTrain = errperc(y,YN,Tolerance);
fprinff(sprinff('                 Percentual Error: %%2.2f %%'),PErrTrain); disp('');

clear x y weight Go G CWNorm GAUSSIAN;

disp(' ');
fprinff(' TESTING...');
%%%%%%%%%%%%%%%%%%% ERROR IN THE TESTING SAMPLES %%%%%%%%%%%%% disp(' ');
fprintf('                 1. Loading Testing Files...'); load testing.dat;
y               = testing(:,10:13); %%%% outputs to the training set.
%%%%%%%%%%%%%%%%% testing dimensions
[N QT] = size(y);
jf (Q ~= QT)
 disp('Error: the testing data is not related to this network - output dimensions do not match!');
 retum;
end;
%%%%%%%%%%%%%%%%% scaling the input vector
disp(' ');  fprinff('                  2. Scaling...');
x = scale(testing(:,2:9),minT,maxT,scales);
[N PT] = size(x);
clear scales testing;
if (PT-= P)
  disp ('Error: testing and training input patterns have different dimensions!');
  retum;
end;
clear QT PT;

%%%%%%%% transforming 'y' into -1 or 1 vectors
disp(' ');
fprinff('      3. Transforming outputs              ');
for i=1 :N
  for j=1 :Q
     if (Y(i,j) == 0)
        y(i,j)=0.1;
else
     y(i,j) = 0.9;
   end;
  end;
end:

%%%%%%%%%%%%%%%%%%% Formating the GAUSSIAN MATRIXES %%%%%%%%%%%%%
disp(' ');
fprinff(                ' 4. Calculating Network Responses...');
GAUSSIAN = ones(N,(sum(Ms)+Q)); %%% (N x (sum(Ms(phase)) + Q biases))
begCent = 1;       %% first column of the vector with the Center positions in each "phase' (1 to Q)
```

```
endCent = 0; %%% last column (idem)
%%% FOR related to the Q Sets of Centers
for phase=1:Q
    endCent = endCent + Ms(1 ,phase);
    %//////////////// CWNorm of the input patterns with respect to the Output coordinate "phase"
    CWNorm = sqrt(P*diag(NORMS(phase,:)));
    %% determines the Gaussian Matrix
    for i=1:N
      if (phase == 1)
        for j=begCent:endCent
           rad = RADIUS(1,j);
           dist = (x(i,:)-CENTERS(:,j)')./sqrt(varT);
           dist = sart(dist*CWNorm'*CWNorm*dist'):
      GAUSSIAN(i,j)=rbfunct(dist,rad,typeRBF);
   end;
else
  for j=begCent:endCent
       rad = RADIUS(1,j);
       dist = (x(i,:)-CENTERS(:,j)')./sqrt(varT);
       dist = sqrt(dist*CWNorm'*CWNorm*dist');
      GAUSSIAN(i,j+phase-1 )=rbfunct(dist,rad,typeRBF);
     end;
   end;
end;

%%%%%%%%%%%%%%%%% Normalized RBF %%%%%%%%%%%%%%
if (typeNetwork)
   for i=1:N
     if (phase == 1 )
%      tot = sum(GAUSSIAN(i,begCent:endCent));
        tot = max(GAUSSIAN(i,begCent:endCent));
        GAUSSIAN(i,begCent:endCent) = GAUSSIAN(i,begCent:endCent)/tot;
      else
         begP = begCent+phase-1; endP = endCent+phase-1;
%     tot = sum(GAUSSIAN(i,begP:endP));
       tot = max(GAUSSIAN(i,begP:endP));
       GAUSSIAN(i,begP:endP) = GAUSSIAN(i,begP:endP)/tot;
       end;
     end;
   end;
  clear tot;
  begCent = begCent + Ms(1,phase);
end; %% end of the phase.

%%%%%%%%%%%%%%%%%%%% Calculating the network answers
%%% FOR related to the Q Sets of Centers
YN         = []; %% network responses (is a N x Q) vector
begCent = 1; %% first column of the vector with the Center positions in each "phaseg" (1 to Q)
endCent = 0; %% last column (idem)
for phase=1:Q
     endCent = endCent + Ms(1 ,phase) +1;
     G = GAUSSIAN(:,begCent:endCent);
     %%% 'weight' is a column vector (Ms(phase) x 1) with the weights of the current set center
     weight = WEIGHT(begCent:endCent,1);
     %%% The network response to the 'phase' output  neuron
    YN = [YN;(G*weight)'];
    begCent = begCent + Ms(1,phase) + 1;
end;

%%%%%%%%%%%%% Formation of the Network response vector
YN = YN': %% is a N x Q vector

%%/ "purification"of the ouput answers (greater than (1-Tolerance) it is 0.9
%%//// and lower than Tolerance, it is = 0.1)
for i=1:N
  for j=1:Q
```

```
    if (YN(i,j) <=Tolerance)
       YN(i,j) = 0.1;
     elseif (YN(i,j) >= (0.9 -Tolerance))
        YN(i,j) = 0.9;
     end;
   end;
end:

%%%%%%% Formation of the last output coordinate %%%
%the last output coordinate is 1 when the others are "off" and 0 otherwise for i=1:N
 YN(i,Q) = 0.9;
 for j=1:Q
    if (YN(i,j) >= 0 5)
       YN(i,Q) = 0.1;
       break;
     end;
   end;
end

disp(' ');
fprinff('      Results:');disp('');
%%%%%%%%%%%%%% CALCULUS OF THE NETWORK TESTING RMS  %%%%%%%%%%%%
Error = y - YN;
RMSTest = sqrt(sum(sum(Error.^2))/(N*Q));
fprinff(sprinff('                        RMSTesting    : %%2.4f'),RMSTest); disp('');

%////////////////// Percentual Error of the Testing
PErrTest = errperc(y,YN,Tolerance);
fprinff(sprinff('                   PercentualError: %%2.2f %%'),PErrTest);

clear i j begCent endCent weight G Go;
%clear CWNorm YN testinq x y phase rad dst;

%%################################
clc: clc:
disp( ==============================================================')
disp(' SUPERVISED (Ort) RBF TO CLASSIFY THE COMPANIES - TRAIN AND TEST' )
disp('=================--===========================================')
fprinff(sprinff('Date                                               : %%s'),date);disp('');
fprinff(sprinff('Expected Number of Centers: %%d'),M);disp('');
fprinff(sprinff('Final Number of Centers         : '));
for phase=1:Q
 fprinff(sprinff('%%d,),Ms (1,phase ))
end;
msg = sprinff(' (%%d subnetworks)');
fprinff(msg,Q); disp('');
fprinff(sprinff('Accuracy in Orthonormalization              : %%2.3f'),PO);disp('');
fprinff(sprinff('RMS Goal              : %%1.3f'),RMSgoal); disp('');
msg = sprinff('Type of Radial Basis Function                : %%s');
fprinff(msg,tRBF); disp(' ');
msg = sprinff('Type of Radial Basis Network   :%%s');
fprinff(msg,tNet); disp('');
msg = sprinff('Method of Center Determination: %%s');
fprinff(msg,tDetCent);
if(typeDetCent)
msg=sprinff(' (P = %%2.2f)');fprinff(msg,PN);
end;
disp('');clear tDetCent;
fprinff(sprinff('Regularization Parameter : %%3.2f (Lambda)'),Lambda);
disp(' ');disp('            TRAINING Results: ');
fprinff(sprinff('            TrainingTime : %%6.4fseconds'),t); disp('');
fprinff(sprinff('            RMSTraining : %%2.4f'),RMSTrain); disp('');
fprinff(sprinff('            Percentual Error: %%2.2f '),PErrTrain);
fprinff('%%');
disp(' ');disp('            TESTING Results: ');
fprinff(sprinff('            RMSTesting : %%2.4f'),RMSTest); disp('');
```

```
fprinff(sprinff('            Percentual Error: %%2.2f '),PErrTest);
fprinff('%%'); disp


function [c,r]=kmeanwn(x,M,CWNorm)
%USAGE [center,spread] = kmeanwn(data,number centers, CWNorm)
%    Inputs: x is (N x p)                          : feature vectors.
%          M                                       : number of clusters.
%    CWNorm is diag(p xp)                          : weighted matrix to obtain norm
% Outputs                                          : cis(Mxp) :clustervectors.
%         r is (M, 1)                              : spread of the cluster c
% Page 247 of "Fundamentals of speech recognition. "
%
[N,p]=size(x);
% xmax=max(max(abs(x)));
% x=x/xmax;     % Normalize the data set.
% c=rand(M,p); % Init. clusters randomly.
c=x(1:M,:);% Init. clusters from first M data pattern.

dist=zeros(N,M);
Chanqe=1; iter=0;             np=zeros(1,N);    xvar=var(x);
%R=cov(x); [V,D]=eig(R); D=diag(D)'; xvar=D;
xvar=ones(N,1 )*xvar;
while Change == 1,
   cp = zeros(M,p);   %%%% cp     - sum of patterns within the center
  cndx=zeros(M,1); %%%% cndx - total of patterns within the center
  for i=1:M,
   dist(:,i)=((((x-ones(N ,1 )*c(i,: (ones(N ,p)*CWNorm) ) .^2) ./xvar)*ones(p,1 );
    end;
[ymin,posCent]=min(dist'); %%% obtain the centers of each pattern (minimum distance)
for i=1:N,
 k=posCent(i);      %%% center k has the minimum distance to pattern x(i)
cp(k,:)=cp(k,:)+x(i,:); %%% adds the pattern x(i) to the others in the center
cndx(k) = cndx(k)+1; %%% increase the number of patterns within the center
end;
for i=1:M,
c(i,:)=cp(i,:)/(cndx(i)+eps); %%% the center is the mean of all patterns in it
end;
if posCent == np,
[nm,i]=min(cndx);
if nm==0,
jj = round(200*rand(1))+1;
 c(i,:)=x(jj,:);
else,
Change=0;
 end;
else,
   np=posCent;
end;
iter = iter +1;
end;
% Compute the cluster spread.
r=ones (M,1);
for i=1:N,
  k=posCent(i);
  r(k)=r(k)+ wnormvar(x(i,:),CWNomm,c(k,:),var(x)))^2; end;
r=r./cndx;
c;    %centers
r;    % spread of centers

function [b] = gramsch(A,g)
%                function [b] = gramsch(A,g)
%  Gran-Schimdt's method
%    Orthogonalization of a vector g into the vector b,
%   according to the basis matrix A.
% INPUTS:
```

```
%    A (N x Ms) - matrix orthogonal
%    where:
%       N - total of vectors in A
%       Ms - dimension of the orthogonal space A
%        g (N x 1 ) - vector to be orthogonalized
%OUTPUT:
%          b (N X 1 ) - vector 9 after orthogonalization
%
[N Ms] =size(A);
[Ng col] = size(g);

if (N ~= Ng)
 disp('Error: incompatible dimensions between matrix 'A' and vector "g"');
 return;
end;
if (col ~= 1)
   disp('Error: vector '9" has to be one-column vector!');
end;

%%% The Gram-Schmidt Orthogonalization process:
%%%-------------------------------------------------------
%%%
%%%The new orthogonal vector is the following linear combination:
%%%
%%% b(i,1) = g(i,1) - SUM ((A(:,i)'*g)/(A(:,i)'*A(:,i)))*A(:,i)
%%%             1,MS
%%% Where:
%%% num = (A(:,i)'*g)
%%% den = (A(:,i)'*A(:,i))
%%% coefs = num/den
%%% ones(N,1 )*coefs' is the expansion of matrix "coefs" to N space
%%%%%%%%%%%%%%%%%% calculus of the coeficients Lambdas
% ///// "numerators"
num = A'*g;             % dimension (Ms x 1)

%//// 'denominators"
den = diag(A'*A);        % dimension (Ms x 1)

%//// "coeficients"
coefs = num./den:       % dimension (Ms x 1)

%%%%%%%%%%% calculus of the orthogonal vector

%/// x = g - (expand matrix of coeficients to dimension N * vector A(j))
b = g - (ones(N,1)*coefs'.*A)*ones(Ms,1);

function [D]=wnormvar(X,C,Y,VAR)
% Weighted equal variance scale: D = sqrt(XC'.C.X'/var)
% weighted in (Haykin, "Neural Networks-A Comprehensive Foundation" pg. 258)
% equal variance scale in (Hartigan, "Clustering Algorithms", 1975, pg. 60).
%
% USAGE
% 1) [D]=wnorm(X,C,Y,VAR)
%             Input: X - input vector (P X 1)
%                : Y - input vector (P x 1)
%                : C - weight Matrix (P x P)
%                : VAR - variance Matrix (1 x P)
%             Output: D - weighted norm of (X - Y)
%
echo off;
[linX colX] = size(X);
[linY coly = size(Y);
[linC colC] = size(C);

if (linX ~=1)1 (linY~=1)
 disp('Error in wnormvar.m: the vectors must be one-line vectors!');    return;
```

```
end;
if (colX -= colY)
  disp('Error in wnormvar.m: the vectors dimensions must be the same!');
  return;
end;
if (colC ~= colY)
  disp('Error in wnormvar.m: C is not a weighted matrix for the vector X!');
end;
if Y==[]
   Dif = X;
 else
   Dif = (X - Y)./sqrt(VAR);
end;
D = snrt(Dif-C'*C*Dif'}:
```

function [r] = Ffunct(dist,radius,typeRB,)

```
% rbtunct(.) calculates the value of a Radial Basis Function
% USAGE: [r] = rbf(dist,radius,typeRBF)
% INPUTS
%         dist (1 xl)-distance     radius (1 x P) -radius
%         typeRBF - identifies which RBF to use
%              1 - Gaussian      2 - Multiquadratic
%              3 - Inverse Multiquadratic     4 - Cubic Spline
%              5 - Linear Spline
%
echo off;
%%%%%%%%%%% Testing the Dimensions:
[linX colX] = size(dist);
[linY coly = size(radius);
if (linX ~=1) | (colX~=1)1 (linY ~=1)1 (colY ~=1)
disp('Error in function rbfunct: the parameters must be numbers!');
retum;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%% Gaussian Function if (typeRBF== 1)
  r = exp(-dist^2/radius^2);
  return;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Multiquadratic Function
if (typeRBF == 2)
   r = sqrt(dist^2/ + radius^2);
   return;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%% Inverse Multiquadratic Function if (typeRBF == 3)
   r = sqrt(dist^2 + radius^2);
   r=1/r;
   retum;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Cubic Spline Function if (typeRBF == 4)
   %% the distance ^3 is the rbf parameter
   r = dist^3;
   return;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Linear Spline Function
if (typeRBF == 5)
   r = dist;
   retum;
end;
```

function [XS] = scale(X,minT,maxT,SCALES)

```
% determines the scaled patteM 'xs' from the raw data 'x'
% USAGE[xs] = scale(x,ScaleMatrix)
% INPUTS:
%         X          is (N x P) - matrix with the input vectors (only Coordinate Values!)
```

```
%              minT         is (1 x P) - vector with min values used in the training
%              maxT         is (1 x P) - vector with max values used in the training
%              SCALES is (N x M) - Scale matrix to determine the case and indexes of scaling
%OUTPUT:
%           XS (N x P) - matrix with the scaled input vectors
%
% Based on Martins & Pacheco's scaling method:
%
% The scaling here always preserves the original interval [min,max]. There have been two cases identified: in one, the
% interval is divided into two parts. The new position of the turning point identified in the coordinate of x describes the
% compression in the original interval. The second scaling case occurs when the interval is divided into three parts. In this
% case, the turning points C1 and C2 are determined according to prior knowledge about the behavior of the coordinate
% of x. E1 and E1 are the scale turning points. They reflect the shrinking or expansion in the original data into the
% scaling interval.
%%          August 1995.
%
[N P] = size(X);
[PS S] = size(SCALES);
if (P -= PS)
   disp('Error in the function "scale": SCALES is not a scale matrix for x');
   return;
end;

%%%%%%%%%%% Adding the Minimum and Maximum of X into SCALES matrix
A = minT'; %% first column is the minimum of each coordinate
A = [A,maxT']; %% second column is the maximum of each coordinate
SCALES = [A,SCALES];

%% SCALES becomes the following matrix:
% columns:
%        1                          2                              3            4     5 6              7
% I min_coord_1_of_x max_coord_2_x scale_case_Coord_1 C1 C2 Comp1 Comp2
% |
% |
% I min_coord_P_of_x max_coord_P_x scale_case_Coord_P C1 C2 Comp1 Comp2 I
%
% where 'Comp1' is the compression grade of the first critical interval
% and 'Comp2' is the compression grade of the second critical interval

%%%%%%%%%% Calculus of the Scaling Turning Points E1 and E2
for i=1:P
 %/// First compression degree is in SCALES(1,6)
 %lil E1 = (c1-min)*(1-Compl/100) + min
 SCALES(i,6) = (SCALES(i,4)-SCALES(i,1))^(1-SCALES(i,6)/100) + SCALES(i,1);
 %/// Second compression degree is in SCALES(1,7)
 if(SCALES(i,3)==2)
    %// E2 =max-(max-c2)*(1-Comp2/100)
  SCALES(i,7) = SCALES(i,2) - (SCALES(i,2)-SCALES(i,5))*(1-SCALES(i,7)/100);
 end;
end:
%%%%%%%%%%% Change in SCALES to accomodate the calculus at once: for j=1:P
    if(SCALES0,3)==1) %% first case at scaling
     SCALESt,5) = SCALES0,2);        % C2 = MAX
SCALESt,7) = SCALES0,2);        % E2 = MAX
   end;
end;

%%%%%%%%%%%%%%%%%% scaling process for i=1:N
  for j=1:P
     MIN = SCALESt,1 );              MAX = SCALESt,2);
     C1 = SCALESt,4); C2 = SCALESt,5);
     E1 = SCALESt,6); E2 = SCALESt,7);
     if (X(i,j) <= C1)
  XS(i,j) = ((X(i,j) - MIN)/(C1 - MIN))*(E1-MIN) + MIN;
     elseif (X(i,j) > C1 ) & (X(i,j) <= C2)
XS(i,j) = ((X(i,j) - C1)/(C2 - C1))*(E2- E1) + E1;
```

```
    elseif (X(i,j) > C2) & (C2 ~= MAX)
XS(i,j) = ((X(i,j) - C2)/(MAX- C2))*(MAX - E2) + E2;
    else
XS(i,j) = ((X(i,j) - C1)/(MAX - C1))*(MAX - E1) + E1;
    end;
  end;
end:
```

TABLE OF CONTENTS

TABLE OF FIGURES